DOCUMENT RESUME

ED 057 663 FL 002 750

AUTHOR	Wang, William S-Y.; And Others
TITLE	Research in Chinese-English Machine Translation. Final Report.
INSTITUTION	California Univ., Berkeley.
SPONS AGENCY	Rome Air Development Center, Griffiss AFB, N.Y.
REPORT NO	RADC-TR-71-211
PUB DATE	Nov 71
NOTE	258p.
EDRS PRICE	MF-$0.65 HC-$9.87
DESCRIPTORS	Algorithms; *Chinese; Comparative Analysis; Computational Linguistics; Context Free Grammar; Contrastive Linguistics; Deep Structure; *English; *Grammar; Language Universals; Linguistic Competence; *Machine Translation; Morphology (Languages); Phrase Structure; Physics; *Programing; Semantics; Surface Structure; Syntax; Transformation Theory (Language)

ABSTRACT

This report documents results of a two-year effort toward the study and investigation of the design of a prototype system for Chinese-English machine translation in the general area of physics. Previous work in Chinese-English machine translation is reviewed. Grammatical considerations in machine translation are discussed and detailed aspects of the Berkeley grammar II and lexicography are considered. Procedures for interlingual transfer are described in detail. The input and output systems, structure and logic of the syntax analysis system, and auxiliary diagnostic processes are covered as are the means for analyzing processed text. A bibliography is provided along with further details on flow charts, documentation, trees, coding, and sentence generation in the appendixes. (VM)

ED057663

RADC-TR-71-211
FINAL REPORT
NOVEMBER 1971

RESEARCH IN CHINESE-ENGLISH MACHINE TRANSLATION

University of California

U.S. DEPARTMENT OF HEALTH, EDUCATION & WELFARE
OFFICE OF EDUCATION
THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL OFFICE OF EDUCATION POSITION OR POLICY.

**Approved for public release;
distribution unlimited.**

Rome Air Development Center
Air Force Systems Command
Griffiss Air Force Base, New York

When US Government drawings, specifications, or other data are used for any purpose other than a definitely related government procurement operation, the government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded, by implication or otherwise, as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

If this copy is not needed, return to IRDT(Z.L.Pankowicz), GAFB, NY 13440.

ED0 57663

RESEARCH IN CHINESE-ENGLISH MACHINE TRANSLATION

William S-Y. Wang
Benjamin K. T'sou
Stephen W. Chen
et al

University of California

**Approved for public release;
distribution unlimited.**

FL002720

FOREWORD

This final technical report was prepared by the University of California, 2222 Piedmont Ave., Berkeley, CA. under Contract F30602-69-C-0055, Job Order 45940000. Authors of the report were Dr. William S-Y Wang, Dr. Benjamin K. T'sou, and Mr. Stephen W. Chan. Other contributors were Messrs. Corey Chow, Herbert Doughty, Robert Gaskins, Steven Huang, Royce Kelley, Robert Krones; Mrs. Sally Y. Lee, Messrs. Charles Li, James Liang, Miss Hasmig Seropian, and Mr. Ronald Sykora. The Rome Air Development Center project engineer was Mr. Zbigniew L. Pankowicz (IRDT).

This technical report has been reviewed by the Office of Information (OI) and is releasable to the National Technical Information Services (NTIS).

Reviewed and approved:

Zbigniew L. Pankowicz

Approved: ZBIGNIEW L. PANKOWICZ
Project Engineer

Approved: FRANZ H. DETTMER
Colonel, USAF
Chief, Intel and Recon Division

FOR THE COMMANDER:
for IRVING J. GABELMAN
Chief, Plans Office

TABLE OF CONTENTS

Page Numbers

Abstract 1

I. Introduction 3

II. The Background in Chinese-English Machine Translation 4

- II.1. Georgetown
- II.2. University of Washington, Bunker-Ramo, University of Texas
- II.3. Peking

III. Grammatical Considerations in MT 7

- III.1. Linguistic Analysis in MT
- III.2. Grammars and Algorithms

IV. Aspects of Berkeley Grammar II 25

- IV.1. Scope of Research
  - IV.1.1. Ambiguities
  - IV.1.2. Complex Sentences
  - IV.1.3. Complementation
  - IV.1.4. Nominalizations
  - IV.1.5. Numerals
- IV.2. Revisions of Berkeley Grammar II
  - IV.2.1. Grammar Codes and Meanings
  - IV.2.2. SVT and Subjectless Expressions

IV.2.3. Ill-formed *R Rules
IV.2.4. Problems Relating to the Lexeme DE
IV.2.4.1. DE and the problem of plurality
IV.2.4.2. The scope of DE
IV.2.5. Sentential Nouns
IV.2.6. Problems with Conjunction
IV.2.7. Passive or Pseudo-Passive Structures
IV.2.8. Copular CT
IV.2.9. Attributive vs. Predicative Adjectives
IV.2.10. Plurality on Verbal Constituents
IV.3. Additional Revisions for Grammar III
IV.3.1. Punctuation
IV.3.2. Features
IV.3.3. Lexical Disambiguation
V. Lexicography 61
V.1. Role of the Dictionary (CHIDIC)
V.1.1. Look-up Phase
V.1.2. Analysis and Parsing Phase
V.1.3. The Interlingual Phase
V.2. Methods of Enhancing Effectiveness of the CHIDIC
VI. INTERLINGUAL TRANSFER 75
VI.1. Linguistic Considerations

        - VI.1.1. The Dictionary
        - VI.1.2. The Parsing Grammar
        - VI.1.3. Form of the Interlingual Rules
        - VI.1.4. Interlingual Implementation
    - VI.2. Technical Considerations
    - VI.3. Structures Requiring Special Interlingual Rules
        - VI.3.1. Insertion of "Be"
        - VI.3.2. Article Insertion
        - VI.3.3. Pronouns
        - VI.3.4. Time and Mood
            - VI.3.4.1. Time Reference in Chinese and English
        - VI.3.5. DE and Noun Compounds
        - VI.3.6. Prepositions
        - VI.3.7. Adverb Shifting
        - VI.3.8. Classifiers

- VII. Programming 95
    - VII.1. The Input System
        - VII.1.1. Chinese Characters
        - VII.1.2. Chicoder
        - VII.1.3. Chinese Teletypewriter
    - VII.2. The Output System
        - VII.2.1. Printed Output
        - VII.2.2. Calcomp Plotted Output

VIII. Structure of Syntax Analysis System (SAS) 100
- VIII.1. General Strategy
  - VIII.1.1. Inputting of Text
  - VIII.1.2. Initialization Phases
    - VIII.1.2.1. Subdictionary selection
    - VIII.1.2.2. Dictionary update
    - VIII.1.2.3. Rule update
    - VIII.1.2.4. Rule adaptation
    - VIII.1.2.5. Dictionary adaptation
  - VIII.1.3. Main Processing Phase
    - VIII.1.3.1. Pre-edit
    - VIII.1.3.2. Look-up
    - VIII.1.3.3. Parsing
    - VIII.1.3.4. Interlingual (UPROOT)
  - VIII.1.4. Output
    - VIII.1.4.1. Printer output
    - VIII.1.4.2. Plotted output

IX. Overview of the Logic of the Present Syntax Analysis System 104
- IX.1. Preparation Phase
- IX.2. Text Processing Phase
- IX.3. Run Termination
- IX.4. Highlights of Software-Hardware Interface

X. Auxiliary Diagnostic Processes 113
- X.1. Updating Data Base

X.2. Concordance
  X.2.1. Rule concordance
  X.2.2. Text concordance
X.3. Random Generation
X.4. Character Plotting
X.5. Break Tables, Constitute Tables

XI. Analysis of Processed Text 116
  XI.1. Run Statistics
  XI.2. Plotted Chinese Trees and Interlingual Trees
    XI.2.1. Chinese Tree
    XI.2.2. Interlingual Tree
    XI.2.3. Ambiguities
  XI.3. Evaluation of English Output

XII. Summary and Conclusions 134

Bibliography 137

Appendix I. Flowcharts 141

Appendix II. SAS Documentation 153

Appendix III. Samples of Plotted Trees 219

Appendix IV. Chicoder Coding (or Decoding) 229

Appendix V. The Sentence Generation Program (SGP) 233

Abstract

This report documents results of a two-year effort toward the study and investigation of the design of a prototype system for Chinese-English machine translation in the general area of physics. Past research efforts at Berkeley had been centered on three areas: (1) contrastive study of Chinese and English, (2) development of an automatic dictionary and (3) programming support for machine implementation.

Our work on grammar (Berkeley Grammar II) in the past two years has focused on the expansion and consolidation of our syntactic rules. Grammar codes in the dictionary and in the rules are reviewed for consistency, and redundancies are eliminated. Further sets of rules are added as a result of the continuing testing and revision of our grammar based on texts in nuclear physics and also on previously existing texts in biochemistry. The statistics on our last run shows that our Syntactic Analysis System (SAS) is able to recognize and parse satisfactorily strings consisting of 20-25 Chinese characters, indicating the ability of the SAS to consistently parse 90% of such sentences.

Testing and implementation of interlingual transfer rules has concentrated on the conversion of Chinese nominalizations and relativizations to their English counterparts by implementing binary permutations and substitution of lexical Chinese-English data. Work is continuing on the implementation of English

complementizers.

Approximately 15,000 Chinese lexical entries, mostly in the area of nuclear physics, have been compiled, coded into standard telegraphic code and added to our dictionary, which now totals approximately 57,000 lexical entries. These entries have been assigned grammar codes, romanization and English gloss.

Programs for the Syntactic Analysis System were written in CDC FORTRAN IV and COMPASS and run entirely on the CDC 6400. In addition to the continual refinement of the managerial, adaptation and parsing programs of the SAS, we have implemented plotting routines on the Calcomp plotter to output structural trees, as well as plotting the corresponding sentences in Chinese characters.

Documentation for all completed programs of the SAS are included in this report.

I. Introduction

The project on machine translation (MT) of Chinese to English at Berkeley has been in continuous existence for ten years. Research during this period may be roughly considered to have progressed in three stages. Phase One concentrated on lexicographical studies. Major research completed during this period (1960 to 1967) has resulted in the compilation of a dictionary (with subject matter mainly in the area of biochemistry). Phase Two initiated syntactic studies near the end of this period with the creation of a grammar for the automatic analysis of Chinese sentence structure.

Phase Three is the integration of the previous two phases leading to interlingual studies concomitant with lexicographical and syntactic research. The specific subject matter is now in the area of nuclear physics. In the Final Technical Report on work accomplished in the contractual period immediately prior to the present one (1967 to 1968), Version I of the Syntactic Analysis System (SAS) was documented. SAS is a package of computer programs capable of accepting coded Chinese sentences as input and producing syntactic trees representing the resulting analysis of the Chinese sentences. Limited interlingual mechanisms were also implemented in these structures.

The present report documents the results of further work in this area during the period of September 1968 through August 1970.

II. The Background in Chinese-English MT

In order to provide a certain perspective regarding the work under report, a brief survey of previous work in Chinese-English MT is presented below. There is no denying the fact that present work has stood on the foundations of earlier efforts in the field and inherited its success and problems. At the same time it also uncovered further problems as well as some new methods of solution consonant with the state of the art.

II.1. Georgetown

The work at Georgetown in the late 1950's was not focused principally on Chinese but was part of a general assault on MT. As was to be expected, initial work concentrated on problems of Chinese text input. Standard telegraphic code was used as the most natural for computer input. The translation process was largely dependent on the result of lexicon look-up. Only a very small sample was used. Sentences had to be processed independently by sophisticated "trial-and-error" procedures, since no appropriate "grammar" was available for this purpose.

II.2. University of Washington, Bunker-Ramo, University of Texas

The work done at the University of Washington under Professor Erwin Reifler was more in the spirit of pre-MT analysis of comparative Chinese-English structure and lexicography. Again the emphasis had to be on the compilation of an adequate glossary of Chinese. Reifler, early in his report [Final Report to the NSF, 1962], pointed out the differences in style, constructions and allowable forms of scientific publications and that no scholarly descriptions of the Chinese language had yet dealt with this aspect of the language. The corpus was not restricted to one field or subfield of knowledge but rather to Chinese scientific texts in general. The choice of such an approach to the corpus was apparently dictated not only by the difficulty of obtaining sufficient material from one subject field at that time but also the hope of gaining "a more representative picture of the general-language problems of the language of science and technology" [NSF Report, pp. 12-13]. As will be seen in our report below, this approach is still premature to a certain extent, and it would have been better to restrict our goal to one field. The effort was concentrated on the study of a refined glossary which would give better word-for-word translations into English. The study produced a glossary of 1880 terms which are of some linguistic value.

The work at Bunker-Ramo was an application of the Fulcrum technique in cooperation with Berkeley and University of Texas. Their work was the development of interlingual mapping and English generation phases of Chinese-English MT by adapting the Berkeley parser and dictionary, representing early efforts of Phase Three of the Berkeley MT system. It should be noted that the partial system implemented by Bunker-Ramo used the SNOBOL3 language, whereas FORTRAN was the mainstay of the Berkeley system and therefore not directly interfaceable. Texas took on the responsibility of expanding and supplementing the Berkeley dictionary. Since 1968 the Berkeley project has also assumed the task of implementing these final phases as well as updating of the dictionary.

II.3. Peking

By 1958 the Linguistic Research Institute of the Chinese Academy of Sciences in Peking has already established a system for the translation of Russian to Chinese. Although the present survey is concerned with work done in the mechanical translation of Chinese as the source language whereas the work in Peking has Chinese as the target language, the fact that mechanization involving Chinese is involved should not be ignored. Their initial investigations took the approach of analysing both languages independently and then attempting translation based on the results of such analysis. They later (1961) changed their approach to that of emphasizing the contrastive analysis aspects even during initial analysis. It should be noted that since their work was Russian to Chinese, many more surface morphological aspects of the source language were available as compared to translation of Chinese to English, in which information on grammatical categories such as plurality, person, noun-verb distinctions, tense-aspect-mood systems, for example, are not well-marked by surface morphology and cause many difficulties in the initial phase of contrastive analysis.

III. Grammatical Considerations in MT

III.1. Linguistic Analysis and MT

Research into experimental MT must be based on a well defined framework of linguistic analysis. The results that have been achieved thus far and the results that can be expected are predictably circumscribed by the particular framework chosen. Moreover, such results are also evidence of the range of limitations and usefulness inherent in the particular theoretical framework. There are important differences between the goals of research in linguistic theory and the goals of research in experimental MT. Although theoretical research is concerned with the totality of linguistic competence, actual instances of such research activities usually focus on particular aspects of this totality. The general approach is that of deduction. Thus, for example, a proposed explanation for complementation in a language in general does not exhaustively take into consideration all of the verbs in the language. Furthermore, in practice, the rules proposed are never exhaustively crosschecked against others in the language. On the other hand research into experimental MT and other activities in computational linguistics must be constantly concerned with the total range of exhaustive application of the results of the more theory-oriented research. Thus inadequacies in the theory oriented descriptions are frequently and constantly unearthed by the more exhaustive concerns of computational linguistics and MT. This underlines the fact that research in MT, not only in theory but also in practice, is concerned with the totality of the descriptive adequacy of the grammar.

Fundamental to all linguistic activities and all other scientific activities is the concern for the basic units in the system. In the case of grammar these units will be the grammatical categories. Grammatical concepts, in the sense of Boas and Sapir, are manifested in the surface structures of sentences in a language, and different languages will have different manifestations of such grammatical concepts.[1] Furthermore, semantically corresponding sentences in different languages may utilize different grammatical concepts and categories, which are realized differently. Numerous scholars have attempted to discuss and outline the nature and range of grammatical concepts. As an example, we may quote Sapir's analysis of "The farmer kills the duckling."

I. Concrete Concept:

1. First subject of discourse: farmer
2. Second subject of discourse: duckling
3. Activity: kill

----analyzable into:

---

[1] See, for example, Boas, Franz. 1970. Introduction to the Handbook of American Indian Languages. (Ed.) Preston Holder University of Nebraska Press, Lincoln; Sapir, Edward. 1921. Language: An Introduction to the Study of Speech. Harcourt, Brace, New York; Jesperson, Otto. 1924. The Philosophy of Modern Grammar. New York; Tesniere, Lucien. 1959. Elements de Syntaxe Structurale. Paris. [14]; and Jakobson, Roman. 1957. "Boas' view of grammatical meaning," in American Anthropologist 61:5, p. 144.

A. Radical concepts:

1. Verb: (to) farm

2. Noun: duck

3. Verb: kill

B. Derivational Concepts:

1. Agentive: expresses by suffix -er

2. Diminutive: expressed by suffix -ling

II. Relational Concepts:

Reference:

1. Definiteness of reference to first subject of discourse: expressed by first the, which has preposed position.

2. Definiteness of reference to second subject of discourse: expressed by second the, which has preposed position.

Modality:

3. Declarative: expressed by sequence of "subject" plus verb; and implied by suffixed -s

Personal relations:

4. Subjectivity of farmer: expressed by position of farmer before kills; and by suffixed -s

5. Objectivity of duckling: expressed by position of duckling after kills

Number:

6. Singularity of first subject of discourse:

expressed by lack of plural suffix in farmer; and by suffix -s in following verb

7. Singularity of second subject of discourse: expressed by lack of plural suffix in duckling

Time:

8. Present: expressed by lack of preterit suffix in verb; and by suffixed -s.

In this short sentence of five words there are expressed, therefore, thirteen distinct concepts, of which three are radical and concrete, two derivational, and eight relational....

Our analysis may seem a little belabored, but only because we are so accustomed to our own well-worn grooves of expression that they have come to be felt as inevitable. Yet destructive analysis of the familiar is the only method of approach to an understanding of fundamentally different modes of expression....

A cursory examination of other languages, near and far, would soon show that some or all of the thirteen concepts that our sentence happens to embody may not only be expressed in different form but that they may be differently grouped among themselves; that some among them may be dispensed with; and the other concepts, not considered worth expressing in English idiom, may be treated as absolutely indispensable to the intelligible rendering of the proposition....

In the Chinese sentence "Man kill duck" which may be looked upon as the practical equivalent of "The man kills the duck," there is by no means present for the Chinese consciousness that childish, halting, empty feeling which we experience in the English translation. The three concrete concepts -- two objects and an action -- are each directly expressed by a monosyllabic word which is at the same time a radical element, the two related concepts -- "subject" and "object" -- are expressed solely by the position of the concrete words before and after the word of action. And that is all. Definiteness or indefiniteness of reference, number, personality as an inherent aspect of the verb, not to speak of gender -- all these are given no expression in the Chinese sentence, which, for all that, is a perfectly adequate communication -- provided, of course, there is that context, that background of mutual understanding that is

> essential to the complete intelligibility of all speech. Nor does this qualification impair our argument, for in the English sentence too we leave unexpressed a large number of ideas which are either taken for granted or which have been developed or are about to be developed in the course of the conversation.... (Language p. 88ff)

On the basis of Sapir's analysis one cannot fail to deduce that perhaps it would be easier to translate from English into Chinese than from Chinese into English. We have already mentioned the research activities in MT at Peking University. It was found that at the level of morphology it is definitely simpler going from Russian to Chinese than in the reverse direction. The subsequent deduction is that the same holds true for English and Chinese MT. This would mean that for the final English output of Chinese to English MT system we would have to devote some efforts toward establishing categories that may be absent in the source language. In the present framework of discussion it would seem that the mechanics of translation may vary under the first approach (pairwise approach). On the one extreme, it could be a word-to-word translation which is quite similar to expressing a message content in the target language using the inventory of grammatical concepts of the source language. Abundant examples of this could be found in the early attempts at experimental machine translation (such as the projects at the University of Washington and Georgetown University). Under the second approach (many language approach), on the other extreme, it would involve several steps: (1) establishing the universal set of grammatical categories (some scholars have associated this with an [intermediate] universal language), (2) establishing the correspondences between the source language and this universal set (or translating into the intermediate language -- attempts at translating natural language into first order predicate calculus may be seen as one attempt in this direction), and (3) establishing the correspondences between the intermediate language and the target language. This will ensure that no grammatical information may be missing and will also facilitate n-tuple interlingual translations. We are not aware of there having been serious and persistent attempts with this approach in mind, beyond the theoretical level.

We can bring our discussion to a more concrete level with reference to the following diagram:

	L_s	L_{t1}	L_{t2}	-------	L_{tn}
A		x	x		
B	x	x	x		
C	x				
D		x			
E	x		x		
F			x		
G					

L_s = source language

L_t = target language

A,B,C,D,E,F,G,... = grammatical concepts

Table 1: "Overt" grammatical catagories in the surface structures of source and target languages

Under the first approach, only L_s and L_{tl} are considered and all features marked for L_s (i.e., B, C, and E) are marked for L_{tl} regardless of whether they are utilized in L_{tl} and regardless of others which may be called into play in L_{tl}. In the second approach, more than one target language may be considered and all features are marked regardless of their utility in the languages concerned. For example, even though G may not be called for in any of the languages concerned, yet it is marked for all the languages concerned.

In an approach to MT that is guided by a practical concern for immediate results, it is necessary to pursue an intermediate course by attempting to analyze L_s in the parsing program with a set of grammatical features (codes) that is the union of the set of grammatical features in Chinese and English. For example, plurality, which is generally not overtly expressed in Chinese, is recognized, while nominal classifiers, which are generally not overtly expressed for non-mass nouns in English, are also recognized. Consider another example based on the distinction between partitive and non-partitive genitive[2] in the two languages under study. The underlying structure for partitive genitive is vastly different from non-partitive genitive. For example, in the following sentences

---

[2] This is sometimes known as subjective genitive and objective genitive in the discussion of Sanskrit and the other languages. Subjective genitive is where the noun in the genitive case is the subject of the underlying sentence, whereas the object of the underlying sentence becomes the noun in the genitive case in objective grammar.

(1) The love of God (partitive genitive)

(2) The fear of God (non-partitive genitive)

(1) is derivable from "God loves X" (X = unspecified object) where *God* is the subject, and in fact we can obtain

(3) the love of God for X (men).

(1) is further related structurally to

(4) God's love (for men).

On the other hand, (2) is derived from a very different source: "X fears God", where *God* is the object and where the subject is unspecified. There is no comparable case for (3):

(5) * The fear of God for X

Instead, it is possible to derive (through nominalization after passivization):

(6) The fear of God by X.

The underlying structure for the non-partitive genitive (2) does not allow still other derivations:

(7) *God's fear for X

and (8) *God's fear by X.[3]

In the case of Chinese there are no non-partitive genitives. The equivalent of (1) is

---

[3] If we would return to examine (1) again, it will be found that (1) is in fact ambiguous, for *God* may also be the object of the underlying sentence: "X loves *God*", which parallels (2).

(1') 神的愛
Shen de ai
God DE love "God's love"

The non-partitive genitive must be expressed as a full clause. Thus the Chinese equivalent for (2) would be:

(2') 人對神的懼怕
Ren dui shen de jupa "The fear men have
Men to God DE fear for God"

These sentences illustrate the crucial difference between the nominalization procedures of the two languages. They further point to the fact that two different Chinese structures are mapped into one kind of surface structure in English. The fact that we can also have (4) in English but not (7) and (8) has no bearing on interlingual considerations for Chinese to English translation, even though it is most significant for reverse considerations.

The central theoretical problem here is, of course, related to the question of whether we can exhaustively enumerate all such concepts, which is in turn linked to questions such as whether meaning can be discretely quantized. At present, linguistic theory and studies in contrastive syntax have not been able to offer us all the necessary insights into language. In fact, because of reasons such as this, many eminent scholars have shunned and disparaged machine translation. Others, underestimating the unsolved problems and basing their hopes on extensive pre- and post-editing, made unreasonable claims with respect to their abilities to provide good translations without linguistic manipulations resulting from serious contrastive studies. Y. R. Chao, when speaking on translation[4], has frequently quoted three requirements set up for translation by a noted Chinese translator, Yen Fu. They are *fidelity*, *fluency*, and *elegance*. He has quite convincingly shown that it is not always possible to fulfill all three requirements with a human translator, let alone with a machine. If we lower our expectations[5] and first concentrate on the problematic areas of the fidelity requirement, we may find ourselves able to garner a recent fruit of technology, the computer, and harness it into performing some simple tasks quickly for us. At the same time, we may be able to shed some light on the nature of the human communication system called language. Viewed in such a perspective, if a machine can render a Chinese *Man kill duck* into an English "Man kill duck", which we can, and more, the practical utility and not futility of this line of human endeavor is already demonstrated.

---

[4] See Y.R. Chao. 1968. *Language and Symbolic System*. Cambridge University Press and Y.R. Chao. 1969. "On Translation", a taped lecture given at UC Berkeley, California.

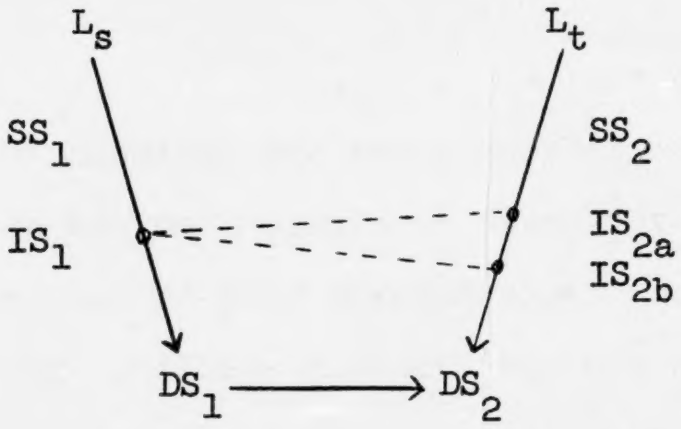
[5] It is interesting to note the recent opinion of an early opponent of MT: "MT research should restrict itself, in my opinion, to the development of language-dependent strategies and follow the general linguistic research only to such a degree as is necessary without losing oneself in utopian hopes." Y. Bar-Hillel 1970. "Position Paper on MT in 1970" Texas Symposium on Machine Translation.

In the past fifteen years the approach to the study of language has been very much revolutionized, and at the same time a much more rigorous framework has been introduced by transformation theory with new and fruitful results. The universal distinction of deep and surface structures has offered new insights into the general structure of language. Recent work in linguistics has raised anew the question of translation.[6] Specialists in the area of second language acquisition have not laid to rest the controversy concerning whether mastery of a second language requires the same process of progression as is required in first language acquisition and whether a second language can be "completely learned", i.e., whether the grammar of the second language will be exactly identical to one that another person has as the first language. Related questions can be posed as to the translation competence of the human interpreter. Consider the following, which is a simplified representation of the linguistic process of translation.

L_s = Source Language
IS = Intermediate Structure
L_t = Target Language
DS = Deep Structure
SS = Surface Structure

---

[6] See Catford, J. D. 1969. A Linguistic Theory of Translation. Oxford University Press, London.
Nida, Eugene. 1964. Toward a Science of Translating. Lieden, E. J. Brill.



Decoding of a linguistic message implies the linguistic competence to relate SS (Surface Structure) and the DS (Deep Structure). The encoding of message and the generation of sentences implies the competence to do the reverse. In fact, linguistic competence may be defined as the ability to relate the two levels of structures in both directions. For the monolingual speaker, his linguistic activities will be altogether confined to a single language. In the case of a translator he has to decode the message (parsing) in the source language (L_s), subject the structural information to interlingual processing, and then recode it (generation) in the target language (L_t). It is not clear to linguists at present whether the translator has to retrieve all the structural information at the level of DS before completing the loop for translation. There is, however, an important distinction between a monolingual speaker and a translator. The message originates in the monolingual speaker and is intimately tied to DS, whereas, on the other hand, the message does not originate in the translator. In the case of a monolingual speaker paraphrasing his own or another person's sentences, there is no a priori reason to think that the amount of information processing reaches into the deepest level of DS. It seems therefore conceivable that the linguistic competence of the bilingual translator could include a certain body of information pertinent to the contrastive differences of the source and the target languages. This body of information by no means makes it altogether unnecessary to resort to analysis at a higher level in DS, for it could be frequently observed that translators do hesitate and pause, and in fact, at times, have to paraphrase. If this is true it would mean that there are Intermediate Structures (IS) which provide a direct input into the interlingual component. Individual input sentences could have Intermediate Structures that represent different levels of completion in parsing.

While the conception of IS is quite clear, the exact nature of IS in an actual context of translation between any pair of L_s and L_t still awaits further research in contrastive studies of the source and target languages. This has been one of the primary concerns of the Berkeley project.

III.2. Grammars and Algorithms

The efficiency with which we can harness the computer in MT is directly dependent on the set or sets of algorithms presented to the machine for implementation. In a most general way, we can understand the "grammar" of a language to be that set of algorithms which describes that language most efficiently. We shall, further, restrict our understanding of the term "grammar" to apply to the syntax of the language only. The grammar then is the set of algorithms or rules which define the syntactic structure of a language. (It should be noted at this point that the terms "grammar" and "language" are not necessarily restricted to our concept of natural language.) They are equally applicable to artificial languages, the languages of algebraic linguistics (context-free and context-sensitive language families), as well as the languages derived from predicate calculus.

Recent linguistic theory looks upon natural language as having the three components: phonology, syntax and semantics. The latter two are the most relevant to our work in MT. The choice of concentrating our attention on syntax does not mean that the semantics of the language is ignored. For MT work, this is in fact an impossibility. The concentration on syntactic work is only a reflection of the state of the art in language theory. Recent work by linguists has shown an increasing blurring of the line between syntax and semantics as evidenced by the works of many others. It seems therefore that as work in MT progresses, more and more attention will be shared between syntax and semantics. However, since the vigorous study of semantics coupled to a syntactic framework is still in the developing stage, an attempt to implement the former effort in

MT work could also only advance in halting steps.

Syntactic study, on the other hand, has already provided many theoretical and practical results which are amenable to computational implementation. We shall single out one specific theoretical treatment,since this has generated the greatest amount of work in the application to both artificial and to natural languages. It is also directly relevant to our present work in MT. We refer to the theory of context-free (CF) languages.

There is no doubt that CF languages are abundantly useful in computer work, since it has been shown that programming languages such as ALGOL and FORTRAN are in fact versions of CF languages.[7] In the area of natural language description,vast amounts of traditional grammatical work on syntactic structure, in fact, turned out to assume some form of CF framework.[8] Therefore, although inadequacies in basing the total grammatical description of a natural language on the CF framework have always been noted, its capability as a vehicle in describing a large portion of the structure of a natural language has never been denied. It has, moreover, been shown that CF languages are equivalent to push-down automata, the latter being one of the basic theoretical concepts which computer scientists have implemented for the efficient manipulation of data structures within the computer.

---

[7] Ginsburg, S. and H. G. Rice. 1962. "Two families of languages related to ALGOL," in JACM 9:3, 350-371.

[8] Postal, Paul. 1964. Constituent Structure: A Study of Contemporary Models of Syntactic Description. Indiana University Research Center in Anthropology, Folklore and Linguistics.

A major consideration in deciding on the form of grammar to use in MT is the ease of implementation. Because of the affinity of programming languages to CF type based languages, the implementation of a grammar of a natural language based on the CF model is extremely attractive.

However, it has already been mentioned that the CF, or, rather, phrase-structure grammar framework has many inadequacies in its ability to handle natural language. As early as 1956, in his "Three Models for the Description of Language",[9] Chomsky had already pointed out some of these inadequacies. Among them was the fact that discontinous constituents, which are so much a part of natural language, could not be handled by phrase-structure grammars. A set of transformations was added to this grammar in order to adequately account for these inadequacies.

There have been some extensions, but still within the CF framework, within MT to suitably account for such discontinuities, such as in the work of Yngve.[10]

---

[9] Chomsky, Noam. 1956. "Three Models for the Description of Language," IRE Transaction On Information Theory Vol. IT-2, Proceedings on the Symposium on Information Theory. September.

The literature on the inadequacies of using a pure CF grammar for natural language description is too well known and extensive, and we shall not pursue this subject in detail here. Be that as it may, this does not mean that work in machine translation must stop and await the complete and satisfactory solution of all the theoretical problems in linguistics. It should be noted that these linguistic issues pertain to the general and universal properties of natural languages. The less than completely spectacular achievements of earlier attempts at MT were aimed at obtaining solutions to very general problems of language, or at obtaining ad hoc solutions to specific problems that could not stand up to exhaustive application. Past experience has indicated that a more restricted goal in MT, taking the state of the art of various related disciplines into consideration, will give a better picture of the work on MT. Bar-Hillel recently stated that "for high-quality MT it is now generally recognized that reliance on the best available linguistic theories is a necessary but by no means sufficient condition," since "even the best modern linguistic theories do not treat adequately the pragmatical [our emphasis] aspects of communication by natural languages."[11] He has redefined "high quality" as a relative term which must take into account the user and the situation. In other words, it is "not inconceivable that a translation program with an output unsatisfactory for a certain user under given conditions might turn out to be more satisfactory if the conditions are changed." Lowering the aim of MT from research on "normalization, canonization or other types of regimentation" to more "language-dependent" strategies has helped to clarify the issues of MT and made it possible for MT workers to attack this problem afresh. In the final analysis the human user in a particular area also is the "first and final judge", and if he is willing to trade quality for speed to a certain degree, then MT has accomplished its task for that user.

---

[10] Yngve, Victor. 1960. "A Model and an Hypothesis for Language Structure," in Proceedings of the American Philosophical Society Vol. 104, #5.

[11] Bar-Hillel. op. cit., p. 1.

IV. Aspects of Berkeley Grammar II

The grammar at present comprises approximately 2100 rules of the context-free phrase structure type. The majority of these rules are of the form

$$A \rightarrow B + C$$

or

$$A \rightarrow D$$

Only a few rules are ternary or quaternary branching. The following sections report on the work in grammatical analysis during the contractual period. The main emphasis in this area has been the refinement and extension of the grammar. The resultant rules reflect a continual process of revision and testing on samples of running text, in particular in the subject area of nuclear physics.

IV.1. Scope of Research

IV.1.1. Ambiguities

One of the significant areas of work is the pruning of ambiguities. On account of the style of individual authors, a written sentence could produce both relevant and irrelevant, as well as spurious ambiguities. The rules of the grammar should preserve and indicate in the final analysis the legitimate ambiguities of each sentence. However, consistent attention has been given to the elimination of spurious ambiguities produced as a result of inconsistency in the grammar rules or inadequate assignment of grammar codes in the dictionary. An example of such ambiguities arising as the result of careful analysis can be seen in the following sentence:

美國物理學會了解脱氫
meiguo wuli xuehui liaojie tuoqing
America(n) physics society understand dehydrogenate(d)

氣體應用的可能性
qiti yingyong de kenengxing
gas(es) application(s) DE possibility (-ies)

may be either:

(1) "The American Physical Society understands the possibilities for application of dehydrogenated gases," or

(2) "The possibility that the American Physical Society understands the applications of dehydrogenated gases . . . ."

美國 物理 學會了 解脱 氫氣體
meiguo wuli xuehuile jietuo qingqiti
America(n) physics has learned extricate(d) hydrogen gas(es

應用的可能性
yingyong de kenengxing
application(s) DE possibility (-ies)

may be translated as either:

(1) "The possibility that American physics has learned the application of extricated hydrogen gases . . ." or

(2) "American physics has learned the possibilities for the application of extricated hydrogen gases."

美國物理學會了解脫氫
meiguo wulixue hui liaojie tuoqing
America(n) physics will understand dehydrogenate(d)

氣體應用的可能性
qiti yingyong de kenengxing
gas(es) application(s) DE possibility (-ies)

may be either:

(1) "The possibility that American physics will understand the applications of dehydrogenated gases . . ." or

(2) "American physics will [emphatic] understand the possibilities for application of dehydrogenated gases."

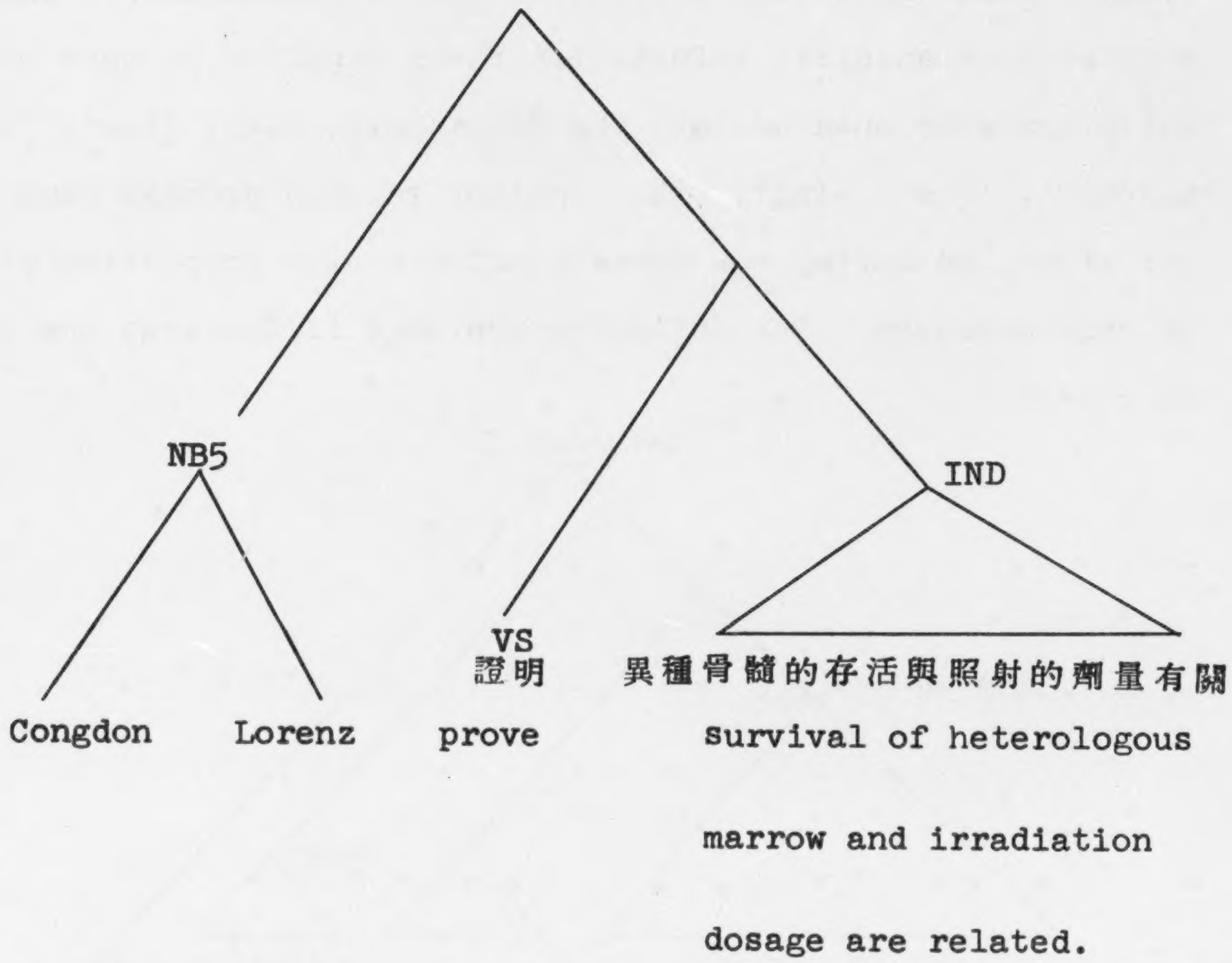
IV.1.2. Complex sentences

Recent statistics on the parsing percentage ability of the grammar has shown that close to 90% of strings consisting of 20 to 25 characters in length are successfully parsed to some higher node such as sentence, clause, noun phrase, or verb phrase. Although relevant sentence length statistics are not generally available, statistics of such a nature would require the accumulated results of a large quantity of machine-processed texts. Determination of sublengths of complex sentences also involves developing syntactic criteria for such segmentations. The Project is accumulating such data as each run is processed. Our corpus indicated that sentences beyond this length are largely complex in nature, that is, longer strings of characters are usually sentences which are either coordinate or subordinate in structure or which contain several levels of embedding.

Among the complex structures that have received the most attention were the sentences with complements and coordinate conjoined structures. Various nominalization phenomena have also required a great deal of study.

IV.1.3. Complementation

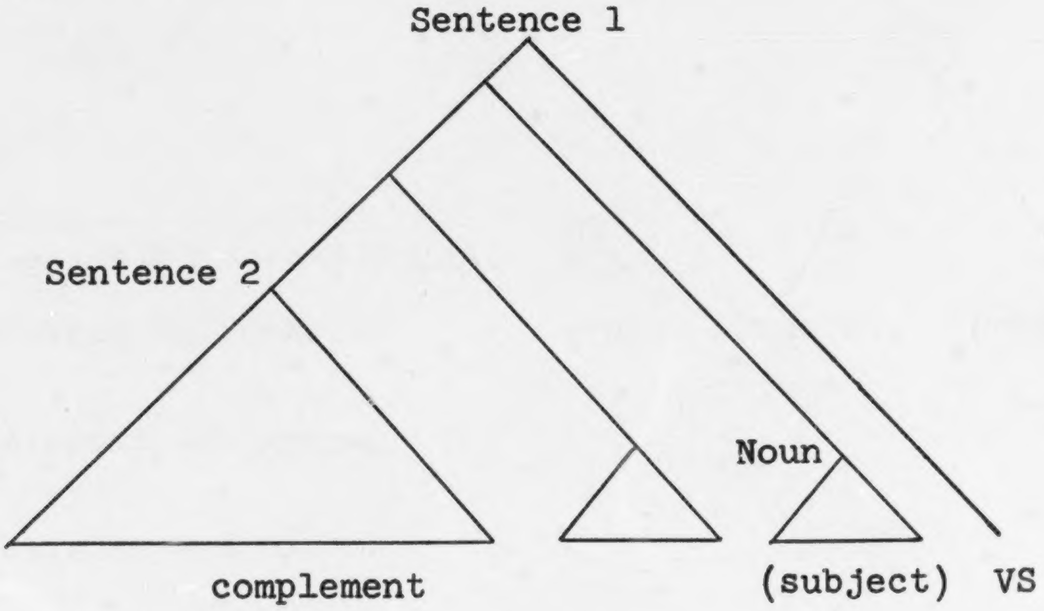
Revision of a section of the dictionary concerning entries listed with the grammar code VS (verbs taking sentences as their object) brought to the surface the problem of accounting for complement structure in Chinese. The VS category proper is the is the set of verbs which take on complements, e.g.,


NB5
IND
VS
證明
異種骨髓的存活與照射的劑量有關
Congdon
Lorenz
prove
survival of heterologous
marrow and irradiation
dosage are related.

The structure designated by IND is the sentential complement of the verb zheng ming 證明 'prove'. This IND itself is a full-blown sentence. At a minimum a rule linking the VS and its complement is required in the grammar (with VI3 indicating this to be a predicate):

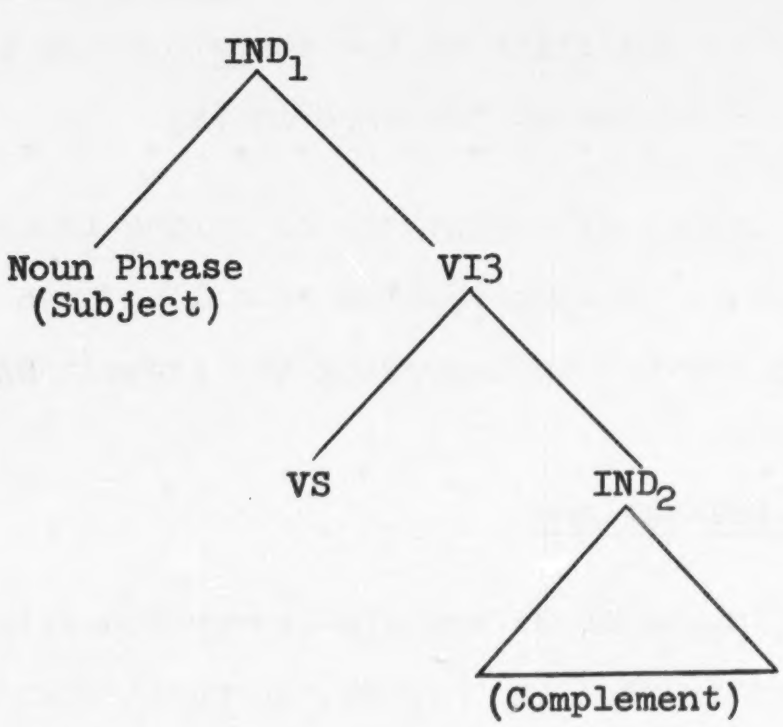
VI3 → VS + IND

However, there are verbs which are not exclusively VS but which may also take sentential complements, i.e., they also act as if they are transitive verbs followed by an object noun phrase (rather than sentential object). Other complementizers which require such specific information to be supplied by each lexical entry are also entered into the dictionary, e.g., 'for', 'to', 'whether', 'V's + -ing', etc. Another related problem came to our attention during our investigation of the properties of the VS verb category. The following sentence illustrates the case in point:

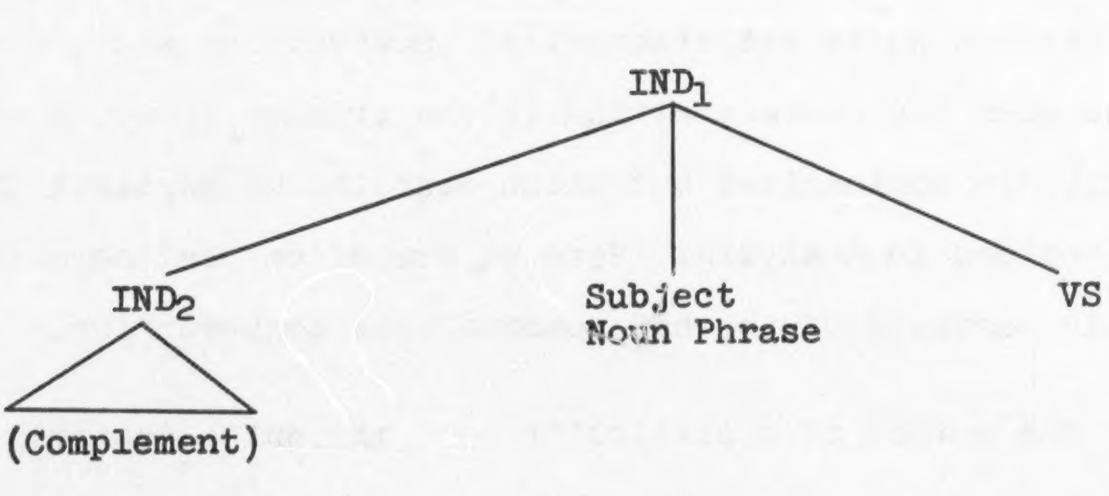


	這種	方法	能	防止	實驗	動物	的	死亡	已	為	很多	學者	證實
[that]	this	method	can	prevent	experimental	animal	DE	death	already	(passive)	many	scholars	verify

The verb 'verify' (zheng shi 證實) at the end of the sentence belongs to the VS category. As such, our rules require that this verb be followed by a complement sentence which does not occur in our example because of the passive construction. In other words, our rules involving VS require the following structure (each IND indicates a type of clause structure):


IND_1
Noun Phrase
(Subject)
VI3
VS
IND_2
(Complement)

whereas our present example has the following structure


IND_1
IND_2
(Complement)
Subject
Noun Phrase
VS

Comparing the two tree schemes, we note that our sample sentence represents a transposition of an entire complement string to the left of the subject noun phrase. Such being the case, we would have to complicate the description of the properties of VS verbs. However, as comparison of the two trees clearly implies, we should be able to preserve the property of VS verbs as verbs followed by complements by applying a string transformation of IND_2 (in (b)) to the right of the VS before the parsing rules apply. We thus arrive at the tree in (a).

This method of making use of string transformations before the actual parsing routine is called into play will be a powerful step forward in improving the present SAS system.

IV.1.4. Nominalizations

The linguistic literature is replete with studies concerning nominalization in Chinese, in particular where an explicit nominalizer DE is present within the construction. This problem has also been dealt with continually in our Project. The present system of rules and the SAS can deal with explicit DE nominalization quite satisfactorily. However, as will often be the case when the surface string is the primary input, a construction which is nominalized but which contains no explicit DE will cause problems in analysis. Here we are often dealing with stylistic variants of the same nominalized construction.

The author of a particular text may quite possibly use

DE in a nominalizing construction in one sentence, and then in the following sentence drop this DE in the same nominalized construction; or even use DE alternately when several nouns occur in sequence as modifiers of the rightmost noun, e.g.,

(a) 放射性 藥液 噴洒 的 方法
radioactive chemical spray DE method

(b) 放射性 藥液 的 噴洒 方法
radioactive chemical DE spray method

(c) 放射性 藥液 的 噴洒 的 方法
radioactive chemical DE spray DE method

It is in fact possible to have the following equivalent nominal where a DE morpheme is inserted between every pair of possible constituents.

(d) 放射性 的 藥液 的 噴洒 的 方法
radioactive DE chemical DE spray DE method

Our present grammar is able to adequately parse construction (d) when all the DE morphemes appear explicitly in the text. However, when the optional use of DE, as in the above examples, is encountered, our grammar would still exhibit ambiguous parsings. It would appear that where such stylistic usage occurs, a probablistic decision has to be made in solving the ambiguities. As our corpus of texts increases we plan to launch a statistical investigation regarding the percentages of insertion and deletion of DE in such constructions.

Not enough is known in the present state of Chinese linguistic studies about the structure of nominal compounds, especially in terms of the transformational derivation of such compounds from more basic structures. Nominalizations in English have received more detailed attention in recent literature [Lees (1960; 1970); Zeno Vendler (1967); Chomsky (1970); Chapin (1967)]. Examples of Chinese nominals which require information regarding "inalienable" are encountered in the AGG (time or locative) type rules where the 'left' and 'right' parts of a compound are split by either a time or locative phrase:

中國近十年來雞蛋很大
Zhongguo jin shi nian lai jidan hen da

In such examples topicalization and/or lexical hierarchial feature relations come into play. (A very interesting study of this problem is found in Bever & Rosenbaum, *Readings in English Transformational Grammar*, 1970, pp. 3-19).

IV.1.5. Numerals

In our present corpus of physics text we have come across quite a number of cases where the sentence could not be parsed due to the inadequacy of rules for handling numerals. We are not only faced with the task of recognizing Arabic numerals as particular constituents, but the Chinese system of numerals is also a problem area. There are two cases that we can consider.

(1) Values of Chinese and Arabic numerals having one-to-one correspondence, e.g.,

Chinese	Arabic
一六二〇 yi lin er ling	1,620
三〇〇五八四 san ling ling wu basi	300,584

(2) Values of Chinese and Arabic numerals not in one-to-one correspondence.

Chinese	Arabic
一千六百二十 yi qian liu bai er shi	1,620 or "one thousand six hundred and twenty"
三十万零五百八十四 san shi wan ling wa bai ba shi si	300,584 or "three hundred thousand five hundred and eighty-four."

Our present grammar rules are able to handle case (1) in a rather straightforward manner, i.e., as one-to-one translations. However our present rules for parsing such strings actually involve ambiguities in interpretation. Observe that in English for the string "1620" we can have either

(a) "one-six-two-zero" or

(b) "one thousand six hundred and twenty"

Our rules recognize the (a) interpretations but would not be able to give an interpretation of the "value" of (b).

We have now revised the rules for parsing numerals. It was found that a considerable number of rules involved only the numeral 1, whereas all other rules of a similar nature involved numerals from 2 upward. The result was that many rules were "duplicated" in this kind of numeral partition. Apparently the logic for the necessity of these two sets of rules came from the fact that information regarding singular and plural was to be captured. Rules with '1' will carry information for singular number and those greater than 1 for plural number.

However, from the systematic point of view the proliferation ("duplication") of such rules does not seem well motivated. Such number agreement information should be provided elsewhere. Only one set of rules for numerals should be present in order to reflect the fact that systematic generalization should be and is possible in the Syntax Analysis System.

In our case (2), translation of Chinese numerals to the corresponding English values are much more complex. Although both languages use the decimal system in expressing units, the values of these units differ in the higher values. Namely, units, tens, hundreds and thousands are strictly equivalent, but 万 *wan* is 'ten thousand', *yi* 億 is 'one hundred thousand'. English lacks both. 'Million' in English is equivalent to 'one hundred *wan*' in Chinese. This means some arithmetic has to be performed in order to arrive at the correct output from Chinese to English.

The other problem is that if the occurrence of *ling* 零 (somewhat like the 'and' in English). The deletion or insertion of *ling* in Chinese still needs further study (Corstius, 1970). It had been suggested that phonetic phenomena may be involved. But there is a possibility that syntactic-semantic criterion may be available. In any case, in our project the problem is not as critical since we are not faced with generating this lexical item, i.e., inserting it in the correct position. In any well-formed Chinese numeral string this entry will already be in the correct place in the string. What is required is then a routine that will recognize this.

IV.2. Revisions of Berkeley Grammar II

In the following sections, rules of the grammar are discussed in detail regarding their function within the system. These represent the more prominent sections in the rules where more systematic revision and/or expansion have been undertaken.

IV.2.1. Grammar Codes and Meanings

Over and above empirical considerations, it is imperative for a grammar such as that embodied in the present Syntax Analysis System to maintain semantically consistent interpretation of each grammatically assigned grammar code. The purpose of this is not merely to insure systematic correlation between grammatical form and meaning, but also to reduce, to a great extent, the burden of meaning rules. With this conception of the organization of grammar, it is for example, logically necessary to delete rules such as the following:

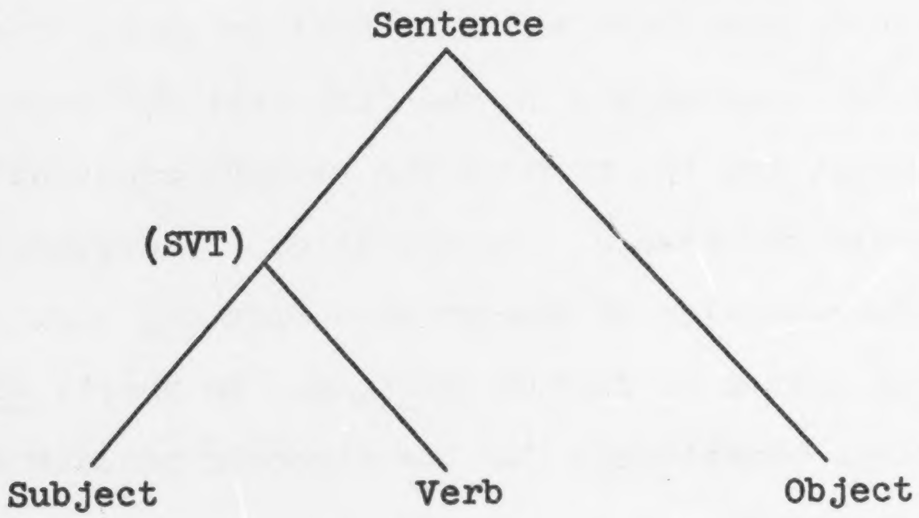
A → VI2
A → VA
A → VI
A → VIN2
A → VIH2
A → VIA2

where A is an adverbial constituent and all the V's are verbal constituents. The assumption here is that a grammar should distinguish those grammatical elements which are adverbial from those which are verbial. Some of the VA's (auxiliary verbs) in the dictionary can be interpreted as adverbial in their syntactic functions, though the converse is not true.

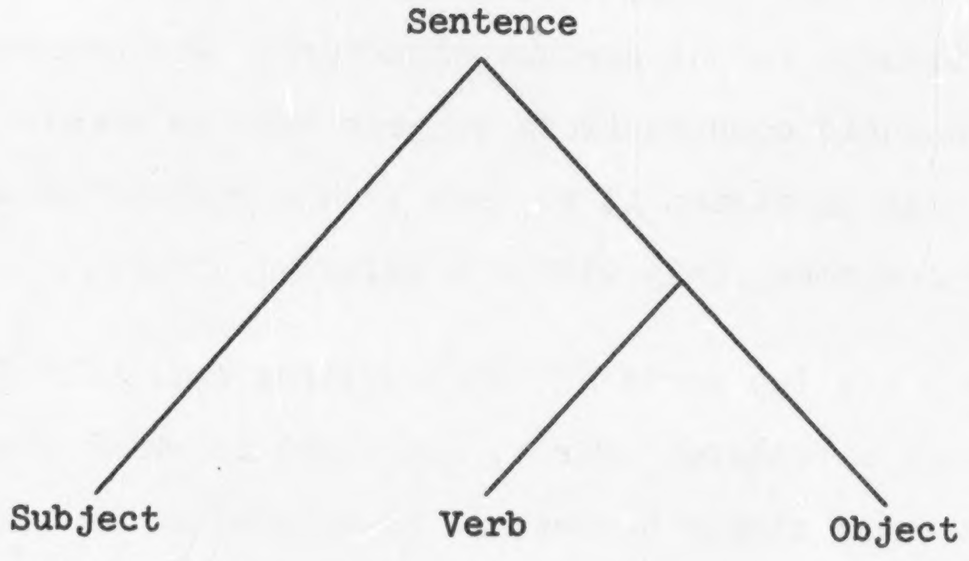
IV.2.2. SVT and Subjectless Expressions

SVT is that grammatical constituent which, by definition, absorbs the subject as part of its underlying structure. The need to identify SVT as a grammatical constituent arises from considerations of the structure of relative clauses in which the subject and the verb form, at the surface structure level, a single constituent in the surface structure. But considerations of other syntactic constructions suggest that we should delete SVT or at least constrain it so that it has restricted applicability, for instance, only within a relative clause.

There are two sorts of constructions for which SVT cannot be the correct structure. First, sentences in which the surface object is deleted either because it is understood or because it can be inferred from some preceding context. In either case, the subject and the verb cannot first concatenate, to which the deleted object would then be adjoined, as if it were an extraneous element, i.e., this means we do not wish to obtain structures of the following type within a sentence:



as opposed to:



In any event, rules which apply only to a constituent formed by the concatenation of the subject and the verb have yet to be demonstrated. Secondly, a grammar which allows for SVT would fail to differentiate it from the passive construction, identified as IND + BE - EN. On the surface level, the passive, much like SVT, consists of a subject and a transitive verb. The difference between the two lies solely in the fact that features associated with the subject and the verb in the passive construction are usually, though not always, incompatible. Consequently, a consistent differentiation of the passive from SVT requires a fairly sophisticated system of feature marking. In short, SVT can't be the appropriate constituent for the elements concatenated in the following rules:

SVT → NA5 + VH
SVT → NA5 + VTHA3
SVT → NA5 + VT3
SVT → NB5 + VT3
SVT → NF5 + VT3
SVT → NH5 + VTHSS
SVT → NH5 + VH
SVT → NH5 + VTH3
SVT → NH5 + VT3
SVT → NXS + VTSS .

IV.2.3. Ill-formed *R rules

Delete

N → NDER*R
N → NDEO*R
N → NDEOS*R
NXS → NDEOS*R

The *R operation is a binary operation which takes two grammatical elements and reverses their surface order. These rules are simply ill-formed and must be deleted.

IV.2.4. Problems Relating to the Lexeme DE

In previous work relating to this project, the treatment of nominalizations involving the lexcial item DE has received much attention (see, example, Ching-yi-Dougherty (1964)).

During the period of our contract, this problem has received further investigation. It should be stated that constructions involving DE (overtly or covertly) constitute a very complex syntactic and semantic problem in Chinese. If one also includes the homophonic and often homographic use of DE, 得 (telecode 1779) and DE 的 (telecode 4104) (the former having the meaning "must", "possible", "should"), the problem is further compounded. In fact, since we are processing printed text, the homgraphy problem is real, although for a specific area such as scientific writing the likelihood of such occurrence is less.

In the above mentioned report, only those cases involving a noun, verb, adjective, noun phrase, verb phrase or complete clause followed by DE and modifying a following noun or noun phrase was studied, i.e., cases where DE is a possessive or relative clause marker.

During the present contract period, the study of DE phrases was extended to sentence final DE and cases where DE is deleted. (See QPR 3, June 1969; QPR 5, October to December 1969; QPR 7, November 1970.)

IV.2.4.1. DE and the Problem of Plurality

There are also rules like the following:

NDEOS*R → NXS + DE
NDESO*R → NXS + DE
NDES → VIS + DE ,

where the intended distinction, if any, between NDEOS*R and NDESO*R fail to come through. Moreover, these rules were also misconceived. Once a noun (NXS) or VIS concatenates with a DE, it is no longer necessary, in fact, incorrect, to mark the resulting constituent to carry a feature of plurality, for the reason that the plurality of that constituent depends not on the noun or VIS but only on the noun following the DE. Whether the grammar is capable of recognizing the number of the following noun is another problem. In (a), 朋友 (peng you 'friend') must be

(a) 他是張三和李四兩人的朋友
Ta shi zhang shan her Li Si liang ren de peng you
He is Zhang-shan and Li-Si two person de friend

(b) X跟Y有相同的地方
X gen y you xiang tung de difang
X and y have similar place

marked as singular, for the subject is singular; in (b) xiang tung is VIX, but it can be either singular or plural, depending in part on the context.

IV.2.4.2. The Scope of DE

Deciding the scope of DE is a difficult problem. In the following rules

NDEP → FNS + DE
NDEP → FN2 + DE
NDEP → NK + DE
NDEP → NM2 + DE
NDEP → NR + DE
NDEP → NRN + DE
NDEP → NRS + DE

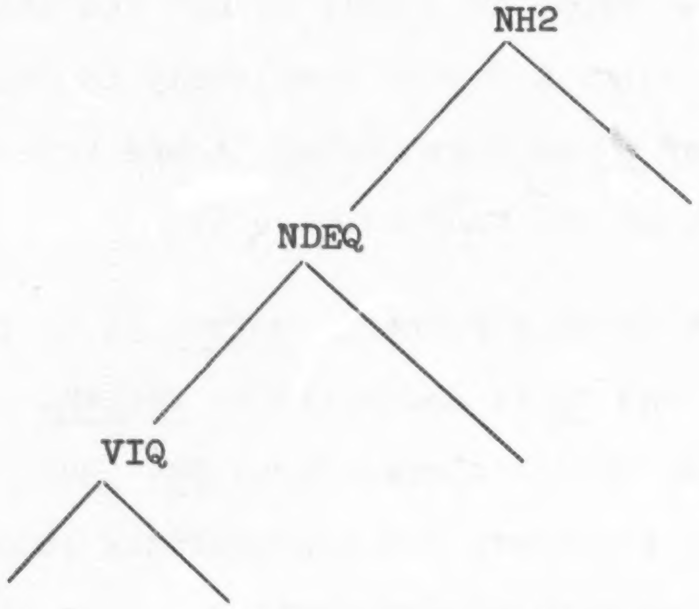
Clause-final DE is not subject to the familiar *R operation, and must be handled, possibly, by disambiguation procedures. But elsewhere the scope of DE has indeterminacy. The above rules imply that regardless of where the DE occurs in a sentence, its scope is always associated with preceding nouns such as kinship nouns, personal pronouns and foreign names, i.e., it is always to be construed as a genitive marker. This, however, is false. In a sentence such as

(a) 遇到張三的親戚
yu dao Zhang shan de qinqi
meet Zhang shan de relative

the scope of DE is either the entire clause, in which case (a) means the relative(s) who met Zhang-san, or just Zhang-san, which is an NM, in which case (a) means (somebody) met the relatives of Zhang-san. The same difficulty applies to VIQ + DE in

(b) 很聰明的人
hen cong-ming de ren
VIQ
very bright de person

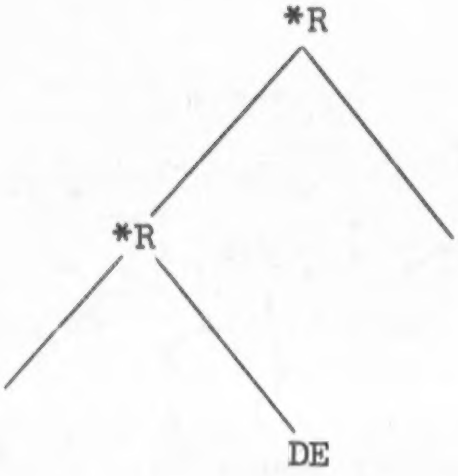
the constituent structure would be



and the *R operation cannot apply, since here DE is part of the modifier phrase. But

(c) 看起來很弱的人
Kan qi lai hen ruo de ren
It seems very weak de person

cannot be assigned the same constituent structure. It would have to look something like



DE takes on the scope of a full clause and the *R operations apply as required. Until a way of completely identifying the appropriate scope of *DE* is forthcoming, these rules can only provide approximations of the real structure.

In VIX → VK + NDER*R, *NDER*R* is to the right of *VK*, the linking verb, and *DE* is dominated by *NDER*R*. Formerly, the rule was designed to handle clause-final *DE*. But, once again, the rule *can* apply elsewhere and consequently loses much of its force, since, as pointed out earlier, we do not yet know just what the scope of *DE* should be in each of the sentences in which it can occur. Similarly, we have to delete the rules:

NDEQ → VIQ + DE
NDEG → AA2 + DE
NDEG → AV2 + DE .

These rules can be called into question on two counts. Built into the rules is the false assumption that the scope of *DE* is always associated with the immediately preceding element. Secondly, adverbs, *a priori*, cannot occur with a *DE* and the resulting constituent cannot be a noun phrase, contrary to what the last two rules indicate.

IV.2.5. *Sentential Nouns*

There is a class of nouns, here called *sentential nouns*, which can take full sentences as their arguments, e.g., *proof*, *theory*, *effect*, *proposal*, etc. It is important to isolate and identify this class of nouns and assign it a distinct grammar code *NBA* since of all the various classes of nouns established in the grammar, only *NBA* can occur in the syntactic environment $\overset{\frown}{\underline{IND}\ \underline{DE}}\!\!\overset{\frown}{\ \underline{N}}$. We also add the following rules:

$$\mathrm{NB5{*}R} \rightarrow \mathrm{DA{*}R} + \mathrm{NBA}$$

$$\mathrm{DA{*}R} \rightarrow \mathrm{IND} + \mathrm{DE}.$$

IV.2.6. *Problems with Conjunction*

One of the many problems connected with conjunction is, of course, the identification of conjuncts associated with the one conjunction marker. This has not been dealt with satisfactorily, partly because the conjuncts could occur in different points in a single sentence separated by other material not relevant to the matter at hand, and partly because the conjuncts may have fairly different surface structures. In the following rules

VIA3 → A + CV + VIA3
IND2 → V2 + CV + IND
VIHA3 → A + CV + VIHA3
VIH3 → A + CV + VIH3
VIH3 → AV2 + CV + VIH3
VI3 → A + CV + VIY
VI3 → A + CV + VI3
VI3 → AV2 + CV + VIY,

inspection of the elements constitutive of the conjuncts indicates that CV, which conjoins two verb phrases, cannot possibly conjoin such disparate materials as A, VIA3 and IND. Work is currently being done on the grammatical structure of conjunction in Chinese.

IV.2.7. Passive or Pseudo-Passive Structures

The following rule has to be deleted:

VIHG → NN5 + VTH2.

Since the features associated with NN5 and VTH2 are different, and since the NN5 precedes the VTH2, by convention they concatenate to become an IND + BE - EN. However NN5 is not necessarily an object noun phrase. There is at least one construction in which the object noun phrase of the verb is preposed to the former case, the preposing of object noun phrases is limited to indefinite nouns and our grammar has provisions for this type of construction.

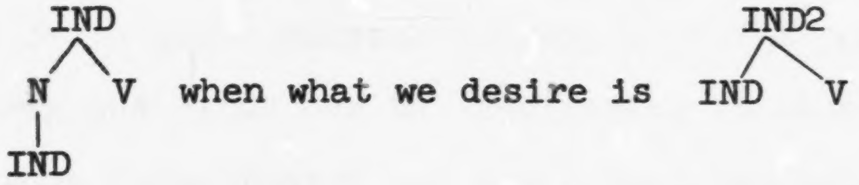
IV.2.8. Copular CT

Add

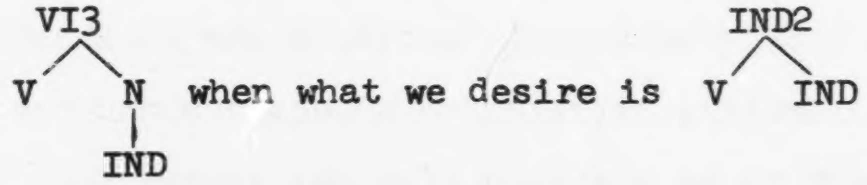
$$\mathrm{IND} \rightarrow \mathrm{N} + \mathrm{VICT}$$
$$\mathrm{VI3} \rightarrow \mathrm{VCT} + \mathrm{VI3}$$
$$\mathrm{N} \rightarrow \mathrm{N} + \mathrm{VI3CT}.$$

VI3CT is a verb phrase constructed from copular CT, which is also a nominal clause introducer, equivalent in function to the whether in English. Thus a noun co-occurring with VICT can be either a regular IND or a noun phrase, hence the additions. Note that concatenation of N with VI3CT is to be done at this level, since we rule out expanding N into IND on the basis of the following considerations:

(A)


IND
N
V
IND
when what we desire is
IND2
IND
V

(B)


VI3
V
N
IND
when what we desire is
IND2
V
IND

IV.2.9. Attributive vs. Predicative Adjectives

Delete

NAS → VQS + NA2
NBS → VQS + NB2
NDS → VQS + ND2
NFS → VQS + NF2
NHS → VQH + NH2
NPS → VWS + NP2
NTS → VQS + NT2
DPS → VWS + DE.

All symbols on the left side of the rule have been replaced by their corresponding second level symbols, e.g.,

NAS → VQS + NA2

become

NAS → VQS + NA2

The existing grammar distinguishes attributive and predicative uses of adjectives (DJ and VQ in the grammar) in ways that are far too inadequate for interlingual conversion. Adjectives in both grammatical functions are indiscriminately marked as VQS. Ideally, attributive adjectives and predicative adjectives ought to be assigned distinct grammatical labels since these two classes of words do not overlap. Within the confines of the category VQS, many can function in either role, which would argue against giving distinct labels to the same lexical morphemes in different syntactic positions. VQS takes plural subjects when used as a predicative. As an attributive, the constituent constructed from the attributive and noun is made, incorrectly, obligatorily plural in the existing rules: NAS, NBS, NDS, etc. This is mistaken, since information on the plurality of the resultant noun phrases depend on this case not on the attributive but on the head noun. These rules would fail generally in cases in which the verbs in question are not marked to take plural subjects. There are only a handful of verbs so marked in CHIDIC and they do not pose any problem as far as we can see.

IV.2.10. Plurality on Verbal Constituents

The vollowing rules have been deleted from our grammar:

VSS → AS + VS
VSSS → AT + VS
VIHS3 → AS + VIH3
VIAS3 → AS + VIA3
VIHAS3 → AS + VIHA3
VINS3 → AS + VIN3
VIS3 → AS + VI3
VTHAS3 → AS + VTHA3
VTHS → AS + VTH3
VTS → AS + VT3
VIASS → AT + VIAS3

VIASS → AT + VQA
VIHASS → AT + VIHAS3
VIHASS → AT + VIHA3
VIHSS → AT + VIHS3
VIHSS → AT + VIH3
VIHSS → AT + VQH
VINSS → AT + VIN3
VISS → AT + VIS3
VISS → AT + VIY
VISS → AT + VI3
VISST → AT + VIT

VIASS → AT + VIA3

Rich proliferation of grammar codes arising from an attempt to differentiate, on incorrect grounds, derived nodes from terminal nodes is demonstrated nicely here as the grammar tries to accommodate such adverbials as AT and AS. AS and AT ordinarily occur with semantically (though not syntactically) plural noun phrases; thus singular noun phrases would be, by virtue of incompatibility of features marked, uncombinable with VSS, VSSS, VIHS3, etc. In the present grammar, marking for plurality on a derived verbal node is redundant, and the rules should thus be deleted.

IV.3. Additional Revisions for Grammar III

IV.3.1. Punctuation

In processing unedited text, a well-developed and systematic set of punctuation symbols generally contributes to correct segmentation of a string into smaller parsable units. In English and other major European languages such a punctuation system is available. For example, the functions of a comma are well-defined and serve to disambiguate a particular string such as the following:

(a) It is clear, however, much of this depends on his ability....

(b) It is clear, however much of this depends on his ability....

Other examples, such as the differentiation between relative and appositive clauses:

(c) The man who owns that dog did not show up.

(d) The man, who owns that dog, did not show up.

are too well-known to enumerate. Punctuation symbols such as the comma and the period are graphic representations of certain pauses in the spoken language, and serve to reduce ambiguity.

Chinese text is less fortunate in not having as well-defined a set of punctuation rules as English. This is a historical problem since in the earlier, classical writings no punctuation was used at all. The present system of punctuation is used in the modern written language: a recent adaptation from the European system. But since the structure of Chinese sentences is quite different from that of, say, English, the use of a borrowed system of punctuation takes on aspects of the functions of both languages. Punctuation symbols under such circumstances become less reliable as syntactic markers in the Chinese text. There is the other aspect of the education of the Chinese author involved. It is quite likely that if his research involves consulting a great deal of English literature, his technical writing in Chinese will be influenced by English punctuation.

In an earlier version of our grammar, commas were incorporated in the rules themselves. Two problems then frequently affected the results of the parsing program:

(1) Parsing was blocked by a comma where the grammar did not expect one, and

(2) Parsing was blocked by the absence of a comma where the grammar demanded one.

In order to remedy these two problems, a new version of the grammar together with a new mode of operation of the system were created so that the parsing system would be insensitive to the presence or absence of commas. This has proved to be effective for sentences of shorter length (for example 30 or 40 characters long). However, for longer sentences which, under these circumstances, would consist of several clauses, the presence of commas at strategic positions within this long sentence would certainly enhance the effectiveness of the parsing. It should be noted that it is not unusual in the written texts to come across strings of 150 characters or more in length before a period is found. These are, in fact, complete paragraphs, within which there occur many simpler clauses separated by commas which may be construed as perfectly well-formed shorter sentences.

However, as was mentioned before, since individual authors are not as consistent in their uses of commas as one would wish, placing too much reliance on punctuation would only bring us back full circle to the original problem. The interaction of punctuation with sentence division within long strings is one area in text processing in which a great deal more study is necessary.

IV.3.2. Features

Linguistic planning of the features necessary to initiate programming was begun near the end of the contractual period. A first step in extracting information from the grammar codes and grammar rule configuration was completed shortly afterwards. The complete reanalysis of the linguistic system will be a major component of future work in this area.

During the latter half of this quarter considerable effort by all members of the staff was devoted to formalizing various approaches to implementing our SAS system using feature matrices. The present system already contains extensive information regarding the combinatorial properties of syntactic categories. This information is represented implicitly in the grammar codes (both terminal and non-terminal). As a next effort in improving the parsing ability of the system, more of this categorical and semantic information will be captured explicitly by using feature matrices. When this task has been completely implemented, it is expected that the simplifications of our present complex system of grammar codes will greatly facilitate internal consistency in the overall system.

Consider, for example, the different noun (N) categories: NA, NB, NC, ND, NAS, NBS, NASS, NBSS, etc. The letters A, B, C, D, S, etc., each represent a particular sub-categorization of the category [A = animate, B = abstract, S = plural, etc.]. These sub-categorizations, whether syntactic, semantic or functional in terms of our implementation, are susceptible of representation by a computer word with bits turned on or off to represent the presence or absence of these features.

The following table is an example of some of the recoding of the present grammar codes into more generalized features which can apply to both nouns and verbs.

Naturally, some of these would be redundant or even impossible. For example, one would hardly expect the occurrence of [+Human] with [-Animate]. As a matter of fact, the features which are extractable from the grammar codes of the present grammar were felt to be inadequate in a full implementation of features, and many refinements are necessary in working out a viable feature system.

	Proposed features				
Existing grammar code	Plural	Human	Animate	Concrete	Some examples
H	0	+	+	+	man, John
HS	+	+	+	+	men
HA	0	0	+	+	run
A	0	-	+	+	bark (dog)
AS	+	-	+	+	
N	0	0	0	+	chair
NS	+	0	0	+	chairs
S	+	0	0	0	they

IV.3.3. Lexical Disambiguation

One of the most persistent problems in resolving ambiguous senses within a particular sentence is that of lexical ambiguity. The lexical item in question has the same grammar code but has several equivalent English meanings. Since the same grammar code is assigned, the rules of the grammar cannot distinguish the different usages. One possibility would be the restriction of the gloss to only one meaning or restriction to one field of knowledge, such as nuclear physics. This would enable us to eliminate extraneous or irrelevant meanings, particularly where nouns are concerned. However, where ambiguity arises, it more often than not involves 'function words', such as prepositions, adverbs, subordinating and coordinating conjunctions. Very few ambiguities arising from these could be accounted for by restriction to a special subject matter since they are not restricted to a special vocabulary. For example, in the phrase

我們下面的報告
women xia-mian de bao-gao
our below DE report

could mean either

(1a) Our report which follows
(1b) Our report below
(1c) in our following report

(2) the report below (underneath) us.

It is more likely, given the fact that 'we' is making the report, that translations (1) are to be preferred. This type of information requires more sophisticated retrieval than can be implemented in our system in the immediate future. Besides this type of information, it is also necessary to distinguish between two very different senses of 'report'. In the first case 'report is used primarily in an 'abstract' or 'non-physical' sense, and secondarily in the concrete, physical sense. In the second case the sense of 'report' is primarily physical, i.e., there is a physical object existing as a report, for example in the form of a set of documents.

Under these conditions the meaning and function of xiamian 下面 must necessarily change depending on whether baogao 報告 ('report') is abstract or concrete within that particular context. Not only is it necessary to have some method of disambiguating this co-occurrence relation in the analysis of the Chinese sentence; this information will also affect the type of transfer rules which are applicable, since, as is clearly seen above, the resulting English output strings assume quite different word ordering.

One initial approach to reducing such lexically specific ambiguities would be to set up a table or list of frequently occurring items which are subject to ambiguous analysis and to assign priority indices to each according to specific environments. In order to achieve practical results these environment checks should be restricted as far as possible to relations within a string or even optimally within its immediate neighborhood. For example in

A.	3 0	度左右	
	3 0	du	zuo you
	3 0	degrees	approximately
	m m	UN	MZ
			AA
			NL

The lexical item *zouyou* has the grammar codes MZ, AA, NL. We wish to have only the grammar code MZ available for analysis rather than allowing all three grammar codes to be passed on to our parsing system. A check to the left of this item shows that UN and MM grammar codes are present. The previously stored table will indicate that the grammar code MZ for *zuoyou* would have the highest rank and therefore pass on MZ as the first alternative. This approach in effect makes use of information supplied by the rules of the grammar even before the string enters the parsing stage, thus filtering out some of the less desirable concatenations. Needless to say, there is no absolute guarantee that the ambiguity will be eliminated when such a technique is implemented on a very general scale. It is necessary that such devices be limited to very well substantiated cases (for example, based on data obtained from concordances) and that such tables be kept reasonably small to achieve practical results.

V. Lexicography

Although there is no denying the fact that the lexicon or dictionary is one of the most vital components in any language processing system, the exact extent to which the dictionary plays a role varies and will depend on the descriptive and analytical machinery which can supply adequate descriptions for a particular sentence in that language.

Only until more recent years has the problem of constructing a dictionary compatible with the grammatical description been given specific attention. Standard published dictionaries, such as Webster's, are "too general". Unless the person is already familiar with the structure of the language and the culture of its speakers, the dictionary does not enable him to arrive at the correct analysis. Since present analysis by machine implies and assumes inadequate or no "knowledge of the world" and only a partially adequate knowledge of the language under analysis, the standard dictionary must be supplemented with a great deal of other pertinent data. For example, it is not sufficient to have the information that a particular entry is a noun, or even an abstract noun. Its relationship with the rest of the sentence and the context is also pertinent data. In a "sentence" such as

The theorem ate bread

it is necessary to have the information for the non-cooccurrence of 'theorem' with the verb 'eat'. This non-cooccurrence can be accounted for in terms of the incompatibility of a particular feature of the noun (namely, [+ abstract]) with the feature on the verb (the latter requiring the [- abstract] or [+ concrete] feature). A consistent approach to treatment of lexical entries using features has been presented in Chomsky [*Aspects of the Theory of Syntax*, 1965].[1] Such a theoretical approach to construction of a lexicon requires long study and only initial steps have been taken. It also assumes that an adequate grammar is also available to take advantage of this framework.

Another area facing the person attempting to construct a dictionary is the problem of compounding and, more basically, the morphological processes of word building. How much information must the lexicon provide to enable the user of the dictionary to recognize the morphological relationships? Is it to list all the derivations fully? No hard and fast rules are available. They must be decided on the particular application for which it is designed.

The Berkeley dictionary is being built in a special area -- nuclear physics. Although effort is concentrated on compiling entries specifically in this field, it does not mean that other entries could be totally ignored since there is a common core of technical scientific vocabulary which is also applicable to this field. One of the problems facing a lexicographer is to decide whether a certain item is a possible entry for nuclear physics or to pass it up as not likely to occur in such texts. Without an adequate number of machine-readable Chinese texts and vocabulary count this situation will not be easily resolved. At present, adequate data on such extensive proportions in regard to the Chinese language is not available.

---

[1] Other more pertinent recent work are those by McCawley, J.D. 1968. "Lexical Insertion." *Papers of the 4th Regional Meeting*, Chicago Linguistic Society; Gruber, Jeffery S. 1967. "Functions of the Lexicon in Formal Descriptive Grammars," Technical Memorandum TM-3770/000/00, System Development Corporation, Santa Monica, Calif.; Lakoff, George. 1965. *On the Nature of Syntactic Irregularity*. Cambridge, Mass., The Computational Laboratory, Harvard University; and Binnick, Robert R. 1968. "On the Nature of the 'Lexical Item'" *Papers from 5th Regional Meeting*, Chicago Linguistic Society, (eds. Daden, B. J., Bailey, C.-J. N., Davidson) University of Chicago, Dept. of Linguistics.

V.1. Role of the Dictionary (CHIDIC)

The various functions of the Berkeley dictionary may be considered under the following 3 phases:

V.1.1. Look-up Phase

For any string submitted to the SAS, the dictionary must be used to (a) correctly segment the string and (b) associate the appropriate grammar codes with each segment, together with the English gloss and the romanization.

Since the sentence is submitted as a string of unedited telecodes the correct segmentations must depend on the existing strings of the dictionary entries to provide the correct match or matches. SAS makes use of the longest match approach in order to facilitate look-up. A maximal string of 7 telecodes is first tested against the input string. When no match is found, a 6 telecode string is tried, and so on. In general, a string of 7 telecodes forming a dictionary entry is relatively rare. Lengths of 2 and 3 telecodes are the most common. The time required for a successful search is therefore quite short using this strategy. Longest match is most practical when we are dealing with a specialized field. For example, contrast the different results obtained by looking up the following 2 string pairs:

(a) 大白鼠 [da bai shu] vs (b) 大 白 鼠 da + [bai + shu]

(a') 小白鼠 [xiao bai shu] (b') 小 白 鼠 xiao + [bai + shu]

	Grammar code	gloss
大 da	VQ	BIG
小 xiao	VQ	SMALL
白 bai	(AA	IN VAIN)
	VQ	WHITE
鼠 shu	NA	RAT or MOUSE

For the longest match the strings (a) would occur as one entry in the dictionary, with the gloss 'white rat' and 'white mouse' and the grammar code NA (noun, animate). For string (b), the resulting look up (by separate telecodes) would be

(b) 'big white rat or mouse'

(b') 'small white rat or mouse'

VQ VQ NA

In fact, we get a wrong match for (b) and (b'). In order to arrive at the correct (b) strings, it is necessary to have the information that da 大 and xiao 小 must first be attached to shu

鼠 in order to obtain 'rat' for *da shu* 大鼠 and 'mouse' for *xiao shu* 小鼠. *Bai* 白 then modifies these two entries. That is, the correct English meaning could be obtained only if we have

白 + 大 + 鼠 bai + da + shu   white + big + rat or mouse
白 + 小 + 鼠 bai + xiao + shu   white + small + rat or mouse

not when we have

大 + 白 + 鼠 da + bai + shu   big + white + rat or mouse
小 + 白 + 鼠 xiao + bai + shu   small + white + rat or mouse.

For such entries as those discussed above it is much more practical to forego the 'building-block' approach and enter these strings in the dictionary without trying to break them down into their component parts. The 'longest-match' principle greatly facilitates the handling of a string which is equivalent to something other than the sum of its parts, or of a string such as *da bai shu*, the meaning of which cannot be obtained by processing its component parts in linear order.

V.1.2. Analysis and Parsing Phase

As soon as the possible correct segments within a string are successfully found, the dictionary will also associate the grammar code(s) for each segment so matched. The parsing program of the SAS makes use of this information in order to parse the string. If each matched segment has more than one grammar code (which is often quite likely), it is not always the case that multiple parsings would arise since this depends on the grammar rules. For example, 研究 yan jiu may function either as an abstract noun (NB) or as a transitive verb (VT) requiring a human or 'organizational' subject 化學 hua xue may also serve as an adjective (DJ), as in

0553	1331	1015	0957	CHEMICAL GROUP
hua	xue	ji	tuan	
化	學	基	團	

Based upon these two examples, there are two entries for each of the above strings in our lexicon. This produces an informational basis whereby all possible choices of functions for each string are available for sentence parsing. One of the tasks which is left to parsing and especially to the grammar is the 'weeding out' or rejecting of those choices which do not match the syntactic and/or semantic restrictions attached to the sentence to be parsed. It may be possible to 'parse' a greater number of sentences with a minimal lexicon in which string functions and attributes are only vaguely specified, but the results of such 'parses' will in most cases be completely irrelevant or even meaningless. Since one of our primary concerns is that parsings be as valid and as correct as possible, the comprehensiveness of the lexicon in general and the completeness of each lexical entry in particular should continue to receive close attention.

V.1.3. The Interlingual Phase

For each segment of the string for which a successful match is found, there is also associated with it the English gloss for that segment. This information is carried along throughout the two previous phases and is available after the English structural tree becomes available. The English gloss field of each dictionary entry actually contains other auxiliary information necessary to obtain the correct English output. For example, the specific environment or context may be included. At present we have also entered information regarding the correct output of English complimentizers depending on the characteristics of the Chinese and English complement-taking verbs. However, in order to make full use of this information further refinement in the present system is necessary. Some of the implications regarding this aspect of development are discussed in the sections on interlingual implementation.

V.2. Methods of Enhancing Effectiveness of the CHIDIC

Ideally, the goal which we seek to attain would be to have a completely comprehensive lexicon based on the structural principles of Chinese and English morphology. In essence this will mean breaking down all strings as far as possible during the stage of acquiring dictionary entries rather than allowing the inclusion of longer strings, which could be broken down into components. For example, in line with this approach, the following strings, having been encountered in text to be processed, would be broken down into their component parts which would form the dictionary entries:

'chemistry research' or 'chemical research'

化 學 研 究 hua xue yan jiu	化學 huaxue	DJ	CHEMICAL
		NB	CHEMISTRY
	研究 yanjiu	NB	RESEARCH
		VTH	RESEARCH*STUDY

'chemistry work' or 'chemical work'

化 學 工 作 hua xue gong zuo	化學 huaxue	DJ	CHEMICAL
		NB	CHEMISTRY
	工作 gongzuo	DJ	OPERATING
		NB	WORK
		VIH	WORK

'radiochemist'

放 射 化 學 家 fang she hua xue jia	放射 fangshe	DC	RADIO
		DJ	EMISSIVE
		DP	RADIOACTIVE
		VT	RADIATE*EMIT
	化學 huaxue	DJ	CHEMICAL
		NB	CHEMISTRY

家 jia	DP	(=NA) DOMESTIC
	DR	(=NK) MY
	NB	FAMILY
	NP	HOME
	NN	HOUSE
	1NB-NH	-IST * -ER * -IAN

With the last example, around 28 different 'parses' are possible. What is needed here is a set of compounding rules which will specify how the given components may acceptably be combined:

[[放 射 + 化學] + 家]
fang she hua xue jia
DC RADIO NB CHEMISTRY 1NB-NH -IST*-ER*-IAN

Also, the order of combination is important:

[放 射 + [化 學 + 家]]
DJ RADIOACTIVE NB CHEMISTRY 1NB-NH -IST*-ER*IAN

= 'radioactive chemist' would be acceptable only in a most unusual and highly unlikely context.

In other words, although a lexicon based on morphological principles would be the ideal and final dictionary, at the present point in time linguistic studies in this area have hardly scratched the surface. (Lu Zhi-wei's Han Yu de Gouci Fa 漢語的構詞法 ('Chinese Morphology') is the only really unified attempt in this direction.) We think that routines such as strong lexical disambiguation procedures and full implementation of a comparatively extensive system of features would enable us to more fully and correctly utilize such a dictionary. Since the development of lexical disambiguation routines and the features system will be part of our next contractual effort, we shall be able to move gradually toward such a more regularized dictionary. However, practicality and efficiency will demand that we utilize the longest match principle for numerous entries which could not currently be handled otherwise.

It has already been mentioned that it would be very desirable to formulate compounding rules or productive morphology rules so as to be able to further refine the dictionary. Deriving such rules would be greatly facilitated if a large dictionary, such as which could be produced by merging entries from CHIDIC, McGraw-Hill, and any other machine-readable dictionaries -- were available, as it is essential in the formulating of compounding or morphological rules to take into account the different functions fulfilled by the same Chinese word or string in different fields. Examples:

2397	1129	(ELECTRONICS) AMPLIFY or AMPLIFICATION
fang	da	(OPTICS) MAGNIFY or MAGNIFICATION
放	大	(PHOTOGRAPHY) ENLARGE or ENLARGING
3564		(GENERAL) SHINE ON or ILLUMINATE
zhao 照		(PHOTOGRAPHY) TO PHOTOGRAPH [as in 照相 'to photograph']

(RADIOLOGY) TO IRRADIATE or TO 'TAKE'
[as in 照X光 to take an X-ray
photograph'] or

IRRADIATION [very commonly used as an
abbreviation of 照射, 'to irradiate'
or 'irradiation']

Or we might note the different English equivalents of DAN XING 單性 0830 1840 in the following entries:

0830 1840 4907 2994 單 性 繁 殖	MONOGENESIS
0830 1840 2702 6792 單 性 核 配	PARTHENOGAMY
0830 1840 5363 單 性 花	UNISEXUAL FLOWER
0830 1840 4814 1395 單 性 結 實	PARTHENOCARPIA
0830 1840 0607 單 性 卵	PARTHENOGENETIC OVUM
0830 1840 3932 2994 單 性 生 殖	PARTHENOGENESIS
0830 1840 1484 單 性 岩	SIMPLE ROCKS

Thus, a broadened program of dictionary acquisition would provide much important information on which the formulation of morphology rules could be based.

Of course, the accumulating of large numbers of dictionary entries is only the first step in the arduous and time-consuming process of adapting such entries to our use within the context of the Syntax Analysis System. Each entry could have to be reviewed and checked for accuracy and applicability, be given a grammar code and feature representation, etc. But the task of refining a large number of 'raw' dictionary entries would be most worthwhile from the standpoint of increasing and improving lexical coverage in more than one field, and also as regards the formulating of morphology and compounding rules.

In the past, CHIDIC has been a static dictionary, one which is somewhat unwieldy and difficult to manipulate in terms of making corrections, additions, and deletions. Various considerations have been under study as to how CHIDIC might be improved, be made more 'dynamic', and be made more readily accessible for purposes of updating and maintenance. For example, it has been generally agreed that some fixed system of record numbering is necessary to simplify the tasks of updating, correcting, and deleting dictionary entries, and that the data structure of the dictionary itself will have to be modified so that feature information and such syntactic and interlingual information as complementizers, prepositions, pointers for discontinuous constituents, and lexical disambiguation tests can be accommodated and, furthermore, be just as amenable to correction and revision as any other information now in the dictionary.

The introduction of features into the lexicon is a significant advance in terms of the entire Syntax Analysis System. Lexicographically speaking, the use of features is a more important step forward than was the earlier adoption of grammar codes, even though a feature representation and a grammar code may in one sense be thought of as conveying the same -- or at least the same kind -- of information. One major advantage of the features system as it relates to the lexicon on the one hand and to the grammar on the other is that it will permit adjustments to the parameters of a singly lexical entry without demanding that the corresponding rule(s) in the grammar be altered, whereas with the old grammar-code system, it was necessary to rewrite the rules of grammar in the event that the lexical item was to be changed or differentiated from other similar but not equivalent lexical items. For instance, if the grammar code of lexical item X were VIHA3 and it was desired that for one reason or another this code needed to be changed to VIHA/NN in order to differentiate lexical item X from items Y and Z, which were similar but not exactly the same, it then became necessary to write a new rule (or more probably, a new set of rules) around the new grammar code VIHA/NN. With feature representations, however, this procedure is no longer necessary; the grammar code -- VI in this case -- remains the same, as do the rules. What may change is the bit representation of the features to be altered.

Summarizing, the Berkeley dictionary has been one of the major accomplishments in having acquired a large data base with which the system could further develop its text processing capability. However, since this data base must work in conjunction with continual developments in syntactic analysis and, as the former task was accomplished at an earlier date than the latter, further refinements to the data base, such as incorporating feature descriptions, will be necessary in order for CHIDIC to keep pace with the rest of the system. Revisions of this nature require detailed study of each entry and are not susceptible to mechanical manipulation. However, the information that is already coded in the present CHIDIC grammar codes can be systematised into feature information to a large extent by mechanical processes, and this will become part of the Project's efforts during the next contractual period.

VI. Interlingual Translation

VI.1. Linguistic Considerations

The interlingual transfer stage may be considered from several aspects:

1. role of the dictionary
2. role of the parsing grammar
3. structure of interlingual rules
4. implementation in the SAS

VI.1.1. The Dictionary

The role played by the dictionary varies with one's concept of the actual structure or content of each entry in the dictionary. In the past, when word-for-word translation was already a viable objective, the dictionary's role was almost overpowering, although in actual fact the dictionary itself was comparatively simple in structure. It was only essential that each lexical entry have an equivalent English gloss. Under this conception, matters such as morphology, syntactic correspondences between two languages, non-equivalent word and constituent order, and so on, could hardly, if at all, play any role on a systematic basis. As a better system of source language parsing in the form of a well-structured grammar was implemented, there arose a greater "sharing of responsibility" between the dictionary and the grammar.

VI.1.2. The Parsing Grammar

The source language sentence must be correctly analysed by the parsing grammar in order to enable the target language sentence to assume the corresponding target-language correct structure, and resultantly to assign the correct semantic interpretation associated with each pair of source-target structures. At this stage the question arises regarding the shape of an MT system grammar. Should the grammar be written in such a way that the resulting structural tree, once it has been parsed by the grammar, directly reflect the structure of the target language structural tree? Or, should there first be a structural tree which shows the source language analysis and then another structural tree showing the structure of the target sentence as the result of interlingual separations on the source structure? Formerly it was thought that since the end product is the target sentence tree, it would seem that the step of producing an intermediate source language structure is extraneous.

We have adopted the second alternative, namely producing both source and target structures based on very practical methodology. The first alternative actually implies that we already have a full-fledged parsing grammar which at all times correctly analyzes the source language sentence and then the rules of the target language are applied to it. It would be difficult to imagine how complex such a grammar would be.

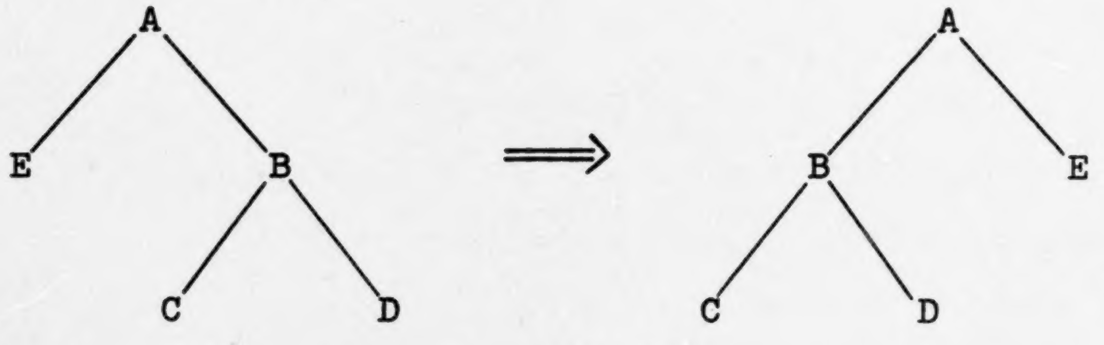
Take the case of the sentences having complements, (or of topicalized sentences) in Chinese. The complement sentence must be first completely recognized as a complement, information on the complement verb must be obtained and the correct complementizers must then be supplied depending on the characteristics of the target complement verb.

The ability to recognize the complement sentence as a sentence already assumes that the grammar has the ability to parse any sentence since the complement sentence is a well-formed sentence. No grammar has yet proved such ability.

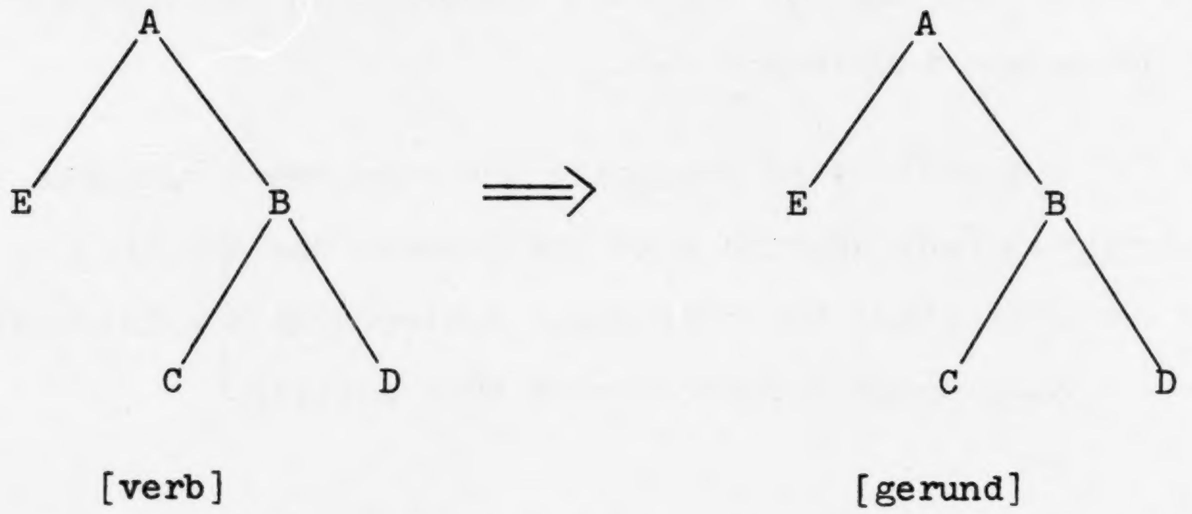
VI.1.3. Form of the Interlingual Rules

Use of interlingual rules is based on the ability to abstract the correct structure or structures from the source language and then operating on these structures. There are at least 3 levels of such rules.

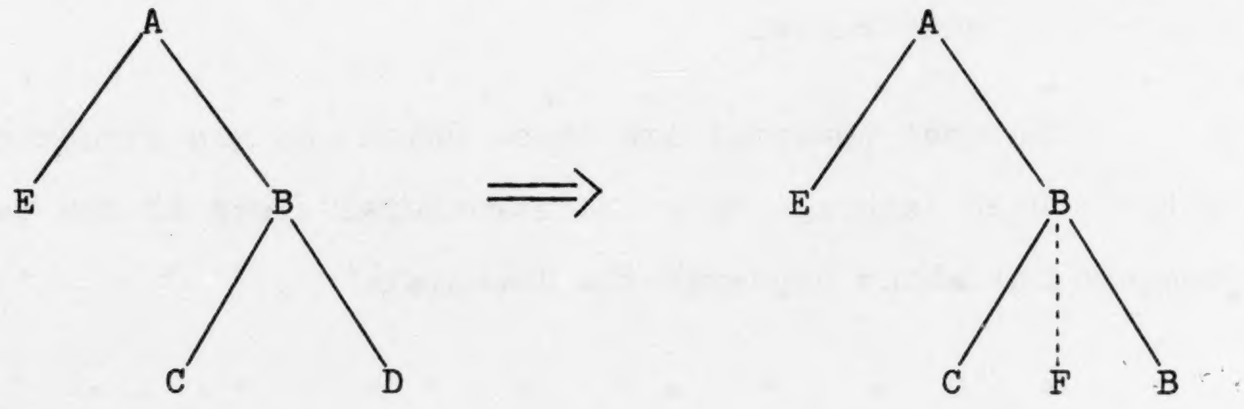
(1) The most powerful are those which can map structural trees of the source language onto the structural trees of the target language but which preserve the labeling:

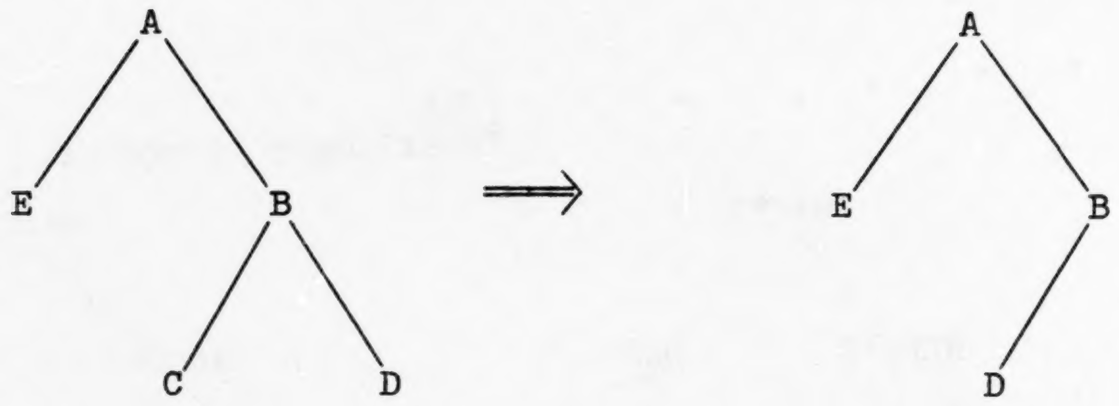


(2) Those which map specific items, resulting in a change of labeling:


A
E
B
C
D
[verb]
A
E
B
C
D
[gerund]

(3) Those which insert or delete items:

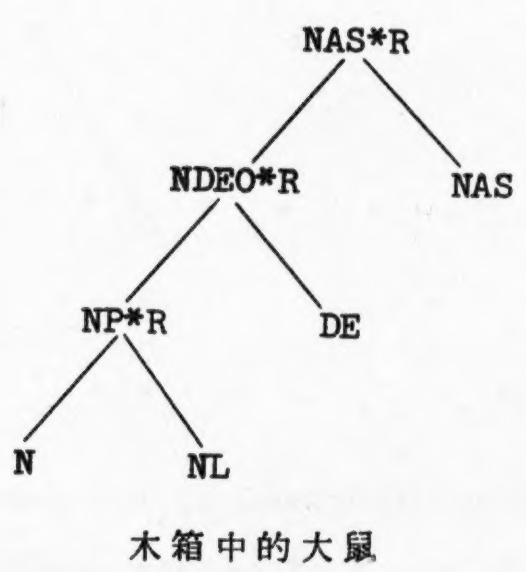

A
E
B
C
D
A
E
B
C
F
B



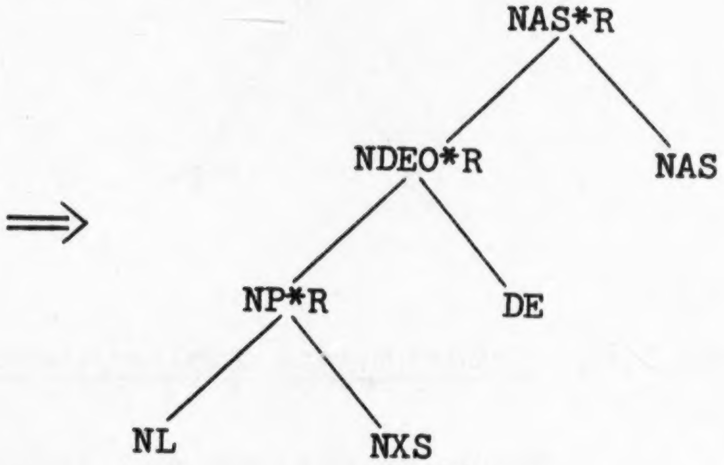
VI.1.4. Interlingual Implementation

The permutations and deletions discussed in the previous section are implemented in the SAS by means of matrix permutation programs which can permute whole constitutes. During the parsing stage the grammar rules involving such permutations automatically assign the resulting node with a tag. The rule representation indicates this by the appropriate node name followed by E.g. *R. When the English equivalent of the Chinese string is required during the UPROOT stage, this tag triggers the appropriate transformation on the constitutes. In the following example, a complex Chinese noun phrase is first analysed, resulting in the Chinese tree (I). A Chinese noun phrase involving the lexical item de requires inversion of the order of the nouns in order to arrive at the order in (IV) to approximate the order of constituents in the English noun phrase.

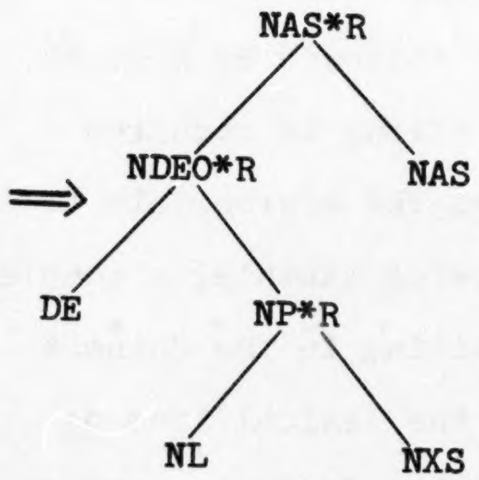
(I)


NAS*R
NDEO*R
NAS
NP*R
DE
N
NL
木箱中的大鼠

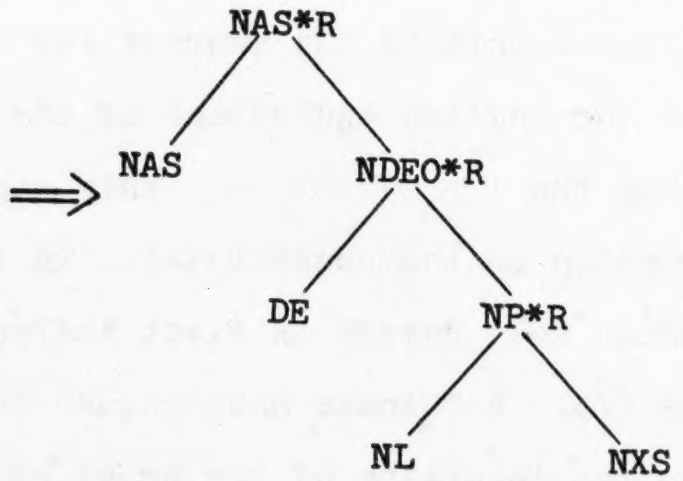
(II)
interlingual transfer


NAS*R
NDEO*R
NAS
NP*R
DE
NL
NXS

(III)


NAS*R
NDEO*R
NAS
DE
NP*R
NL
NXS

(IV)


NAS*R
NAS
NDEO*R
DE
NP*R
NL
NXS

It will then be necessary in the English morphology and syntax to make adjustments to the above structure to obtain the English output. For example, the lexical item de has to be deleted and the English definite article the has to be inserted between in and wooden box as well as preceding mouse. These are by no means trivial tasks. A great deal more study in the area of Chinese-English contrastive studies is required to determine such correspondences.

It should be noted that in the transformation of the tree structures from (I) to (IV) information for each permutation was obtainable by inspecting the immediately dominating node. It was not necessary to go down further to check the constitutes dominated by this particular node. However, the case for obtaining a reasonable English string is more complex. In general, it will be necessary to examine the terminal grammar code and even the English gloss. For example, NL is the grammar code assigned to the Chinese entry zhong 中 , which has English glosses in, inside, within, middle, midst. The correct preposition has to be chosen with respect to the context. This means at the least that the following noun must be examined. This latter grammar code NXS is itself a complex unit indicating plurality of the noun or noun phrase. This means that another procedure must be called in to perform a different set of adjustments. As a matter of fact, in the present example, plurality is also indicated in the grammar code NAS. Thus mouse must be changed to mice. This last process should of course be a very late step in refinement. At the present stage of research one probably need not implement such a step, which is recognized by everyone as being highly idiosyncratic in the treatment of the English language.

VI.2. Technical Considerations

In line with the foregoing discussion, the parsing of sentences in the source language must capture enough information to make generation of corresponding sentences in the target language possible. This implies that words which function the same in the source language, but whose translations function differently in the target language, must be assigned different grammar codes. Similarly, nonterminal syntactic units with distinct translations must receive distinct grammar codes in spite of similarities in source language function. These considerations are a crucial part of the writing of the rules by which the source language is parsed. So even total separation of the application of interlingual transformations from source language parsing cannot provide a way to apply meaningfully contrasting alternative systems of interlingual transformations to a given parsing of a given sentence. To correct an inadequacy in translation, it is usually necessary to subdivide grammatical categories, to provide corresponding rules for the new categories, and to reparse the sentence before applying any corrected system of interlingual transformations.

Decisions as to which interlingual transformations (whether permutations, insertions, deletions, or combinations of the three) are to be applied to which type of node are made while writing the rules and setting up a system of types of node. It is therefore reasonable that each grammar code or type of node should at this time be marked for the application of whatever transformation is deemed appropriate. For this reason, the name of each grammar code intended to receive transformation, "T", is given the ending "*T". For example, NB4*R, NN4*R, NDEO*R and many other grammar codes are marked to receive the transformation "R" or reversal, which is the binary permutation. It is the ending which determines which transformation is applied to which node, so it is unfeasable once a sentence is parsed to apply contrasting systems of interlingual transformations.

The logical integrity of the process of applying interlingual transformations may be maintained equally well whether the transformation subroutine is called from the parsing routine after each constitute is formed, or whether it is called after the entire sentence is parsed and applied in turn to each constitute in the table. The only thing to be gained by applying the interlingual transformations to the constitutes as soon as they are generated is the preservation of information about the position in the sentence of the left end of the constitute. This information could be lost since it is not contained in the constitute itself, but only in the position of the constitute within the table. This loss of information is not serious, however, since the lost information is not used by the printout program and would not be used by any future printout programs now envisioned.

The considerations previously thought to be important in the question of separating transformation from parsing have now been shown to be of small consequence. The desirability of separate printed trees for both languages seems far more relevant, for it now appears that we can apply the interlingual transformations after the entire sentence has been parsed.

In fact, the question may now be raised as to whether to call the interlingual subroutine from the second pass of the printout subroutine. The elimination of this step would have the advantage of saving machine time since the transformations would only be applied to those constitutes actually relevant to the parsing of the sentence.

Much effort can be saved by applying permutations to node associated with long rules. For example, the transformation "X" is the permutation $\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 5 & 1 & 4 \end{pmatrix}$ which is appropriate to the relative clause construction:

$$\text{IND*X} \longrightarrow \text{NX5} + \text{SHIR} + \text{AA} + \text{V} + \text{DER}$$

for which the English order should be:

$$\text{SHIR} + \text{AA} + \text{DER} + \text{NX5} + \text{V}$$

The application of permutations to long rules is somewhat complicated by the fact that for $n > 2$ each n-ary rule is represented in the machine by a string of $n-1$ binary rules. The procedure by which this obstacle is being overcome may be illustrated by the following example:

The above rule

IND*X ⟶ NX5 + SHIR + AA + V + DER

is represented in the machine by the four binary rules:

IND*X → NX5 + 4;y

4;y → SHIR + 4;y

4;x → AA + 4;w

4;w → V + DER

where the grammatical categories 4;x, 4;y, and 4;w are manufactured by the machine for the representation of this rule only and can occur in no other rules.

The constitute table representation of the corresponding nodes for an instance of this rule in a particular sentence could be, for example:

ADDRESS	L	R	G	P
1277	1272	1133	A(IND*X)	
1272			A(NX5)	
1133	1132	1017	A(4;y)	
1132			A(SHIR)	
1017	1015	721	A(4;x)	
1015			A(AA)	
721	717	403	A(4;w)	
717			A(V)	
403			A(DER)	

where L = address of left constituent

R = address of right constituent

G = address of grammar code

P = sentence position of right sibling

and A(GC) is the relative address of grammar code GC.

The important thing to note is that it is necessary to permute just the addresses underlined, only the last of which is not in an L field. To accomplish this a temporary dummy constitute, 9000, is formed with 403 in its L field. Then just the L fields of constitutes 1277, 1133, 1017, 721 and 9000 are permuted and the L field of the dummy is returned to the R field of 721.

VI.3. Structures Requiring Special Interlingual Rules

The following subsections deal with areas in Chinese which require specific treatment in the English target language in order to obtain acceptable output. The conclusions were based on comparisons with the Chinese text with different stages of English translation. The translations were at first free in order to see the differences and evaluate the problems at maximum. Gradually, however, the translations were controlled in order to identify the minimum necessary changes to produce not the best but at least the most faithful and comprehensible English translation. The comparisons were carried out mainly by drawing the structural trees for the Chinese and the corresponding English sentences; deletions from Chinese and all additions necessary for English were marked.

VI.3.1. Insertion of 'Be'

The absence of a linking verb be in Chinese and its presence in English makes it necessary to insert be wherever necessary. The terminal grammar codes (identified so far) that require be in English are VQ (the stative verbs), VI02 ('weather verbs'). Also, in long relative clauses without a verb, it is necessary to have a be in English.

VI.3.2. Article Insertion

The definite ('the') and the indefinite (a/an) articles are the most frequent additions necessary for English. Article insertion has always been tackled in previous MT work with varying degrees of success. It still remains as an extremely complicated task. However, we are at the point where a first attempt can be made to insert these articles mechanically by applying some of the formalism developed recently in discourse analysis.

VI.3.3. Pronouns

The addition of pronouns and the agreement of person and number between pronoun and verb requires further work. It is possible that recent developments in discourse analysis will be of great relevance.

VI.3.4. Time and Mood

The problem of time and mood as controlling the tense and aspect in the verb. The time expressions in Chinese and the aspect system of the Chinese verb are under study at present.

VI.3.4.1. Time Reference in Chinese and English

A detailed study of 230 time words (about half collected from CHIDIC) was undertaken to compare and contrast the syntactic and semantic properties of time words in Chinese and English. Time reference in English is introduced by different prepositions with relative freedom of word order, depending on duration reference or unit reference (more commonly referred to as point reference). However, time reference in Chinese, regardless of duration or unit reference, usually requires different or no prepositions and the requirements of word order are different from English. For example,

(1) He came for an hour

(2) ta lai guo yi xiau shi

他來過一小時

(3) He came in an hour

(4) ta zai yi xiau shi nei dao le

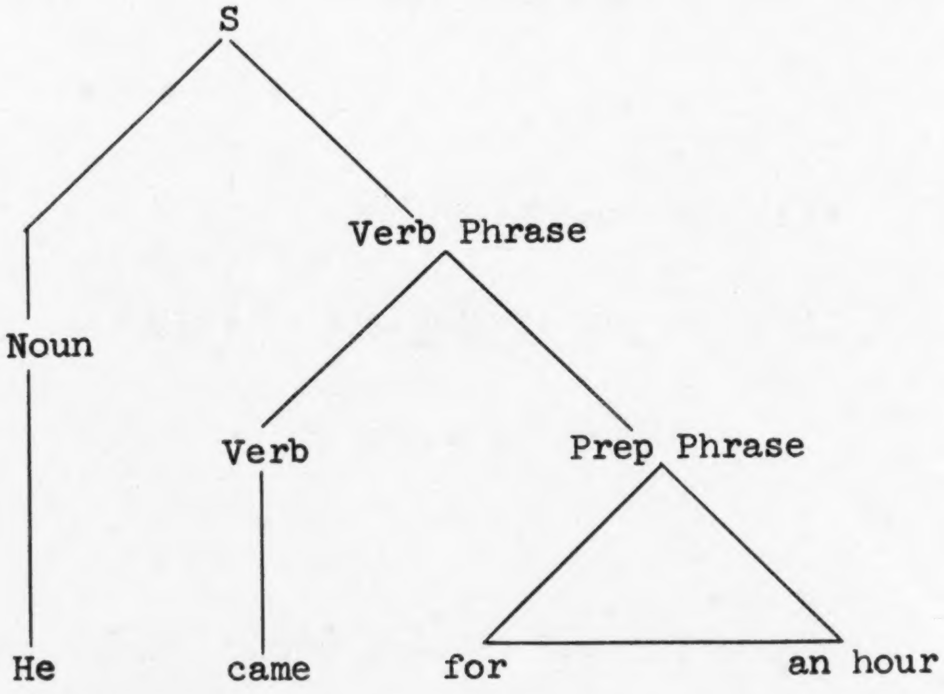
他在一小時内到了

Sentence (1) and (3) are translations. The Chinese sentence has no preposition when the time refers to a continuous period of duration. That is, at any time during that hour, he was present here. Sentences (2) and (4) refer to a period of time which cannot be subdivided. That is, for any sub-period of time within that specific hour he had not yet succeeded in arriving here. The Chinese sentence requires a totally different structure, viz., the discontinuous constitutes zai.....nei, which specifies the totally, enclosed time span: the 'within' duration.

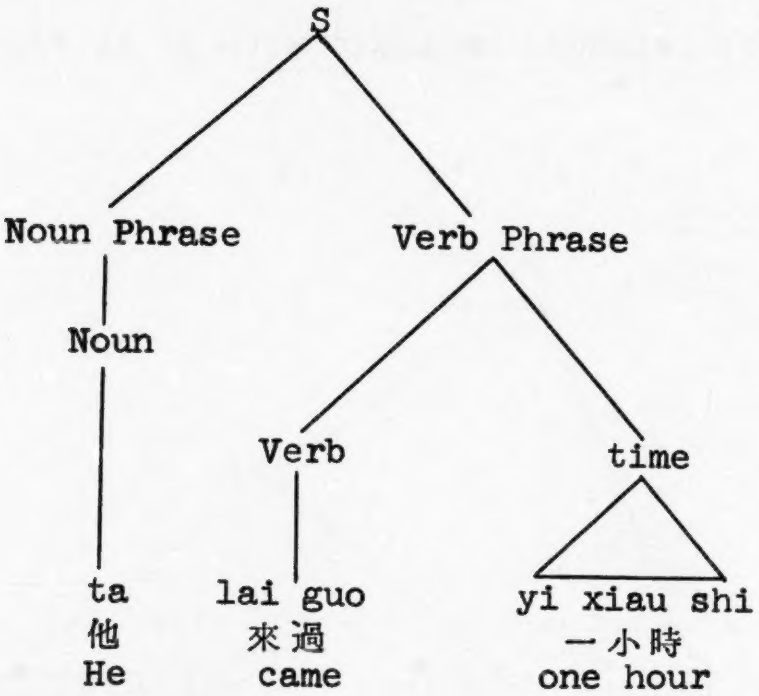
(The reader is referred to Vendler [1967] for further observations on the English case.)

The structures of the English and Chinese sentences are quite different, as can be seen in the following diagrams, in particular for sentence (4) where the time phrase must precede the verb.

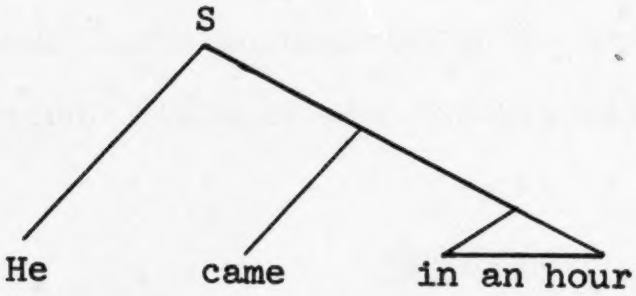
(1)


S
Verb Phrase
Noun
Verb
Prep Phrase
He
came
for
an hour

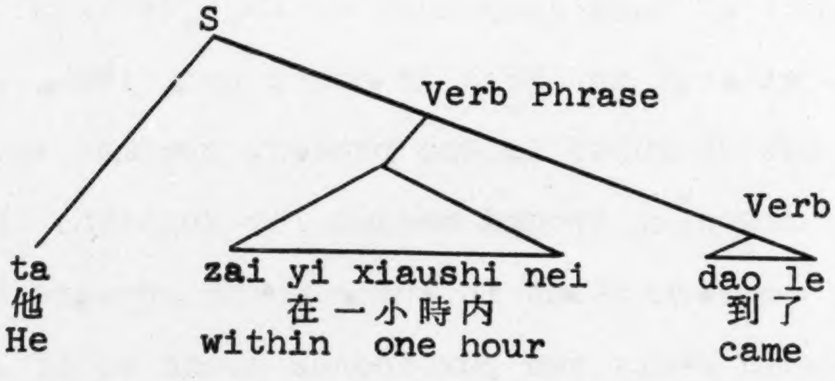
(2)


S
Noun Phrase
Verb Phrase
Noun
Verb
time
ta
他
He
lai guo
來過
came
yi xiau shi
一小時
one hour

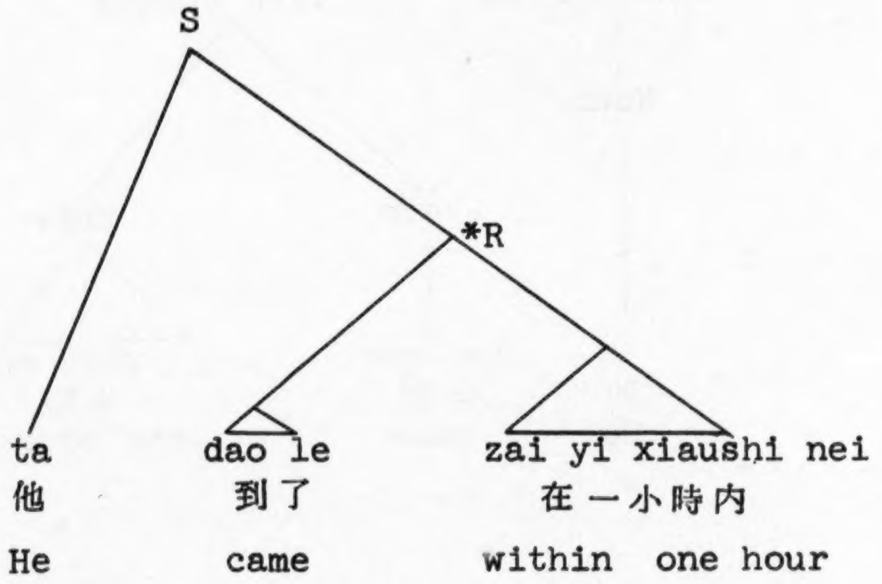
(3)


S
He
came
in an hour

(4)


S
Verb Phrase
Verb
ta
他
He
zai yi xiaushi nei
在一小時内
within one hour
dao le
到了
came

To arrive at the corresponding English structure, (4) must be subjected to an interlingual transfer rule, similar to that suggested for compound nominals with de as represented by the diagram:



The handling of discontinuous constitutes in the time phrase in (4) is non-trivial and is still under study.

VI.3.5. DE and Noun Compounds

By far one of the most persistent problems we are faced with is that of noun compounds or the juxtaposition of more than two nouns without any DE's in overt positions. Wherever DE appears, the *R rules in the present grammar take care of the necessary flipping around needed for English. One possible solution for constructions in which no DE appears has been to add DE in between every two juxtaposed nouns or if at least one DE occurs in the sequence. However, this is much too ad hoc and will lead to just as many ambiguities as before (if not more). A concordance and statistics on these modifying structures in Chinese and English will be necessary to make the decision of first preferences and default options easier.

VI.3.6. Prepositions

The problem of prepositions idiosyncratically required by individual verbs (e.g., 'be interested in') seems to be best handled if these prepositions are made optional parts of the English glosses in CHIDIC. Special routines will later delete prepositions that are juxtaposed as the result of interlingual transfer.

The differentiation into conjunctions, comparisons, and complementations of compound sentences which have no direct indication of which is meant was another major problem in our work. First, a revision of the entries of verbs that require sentential complement (the VS category) was found necessary. The revision of these entries was carried out with the requirements for these structures in English in mind. Now that that task is completed, we think we have a reasonable framework with which to investigate sentences with sentential complements.

VI.3.7. Adverb Shifting

That certain adverbs have to be in certain positions in

English sentences is agreed upon. But whether an adverb shifting rule which places most adverbs preceding sentences in Chinese at the end of sentences in English is still under consideration.

VI.3.8. Classifiers

Classifiers in Chinese is another classical problem since although English in general does not require such equivalents, some cases do necessarily require them. For example, Chinese yi zhang zuozu and yi zhang zhi are translated as a table and a piece of paper respectively. The classifier zhang is consistently used for the Chinese case.

The preceding section mentions only some of the more recurrent areas of investigation in the course of work on inter-lingual transfer analysis. These are very specific problems requiring down to earth solutions in order to facilitate the correct translation of the source language into the target language.

VII. Programming

VII.1. The Input System

VII.1.1. Chinese Characters

The unique system of Chinese character writing has posed, from the beginning of automatic language data processing, problems of efficient input of text for a machine which accepts alphanumeric symbols. The large number of homophonous words in Chinese presents great problems of ambiguity resolution if an input text is coded by means of any of the romanization systems in use today.

Standard Chinese telegraphic code has been the method of coding which provided a unique, one-to-one representation for each character. Each character is coded into a four-symbol (generally four-digit) code. In terms of computer storage and manipulation of data the fixed length coding for each character is very efficient.

Its drawbacks are that the coding process is much more tedious and more prone to human error. Besides, once a text has been encoded, the text is no longer intelligible except to someone very proficient in reading telecodes. Ambiguities do arise in the codes themselves because of changes and rearrangements, by the addition of characters not originally included, etc.

VII.1.2. Chicoder

The Project has investigated alternatives for speeding text input. The device known as the Chicoder has been used by our project. It encodes Chinese characters from keyboard input on punched paper tape, which could in turn be converted to computer readable magnetic tape. By hitting two keys on a typewriter-like keyboard a five-by-five matrix of characters is shown on a screen. Pressing a third key will select the desired character once the correct matrix has been called up.

The two keys which locate the matrix represent the upper or left part of the character and the lower or right part. It is not always possible to locate this matrix on the first few tries.

Although the machine was originally designed with a printer to produce hard-copy for proof-reading, that feature was not available on the present machine. It is extremely difficult to correct errors under such circumstances.

When the Chicoder was delivered to us, it was not in operation order. We have successfully brought it back into operational mode again.

A program has been written to convert the coded paper tape to our telegraphic code so as to interface it with our SAS. We are still exploring this area of interface, but we have come across instances in which ambiguous or incompatible codes have been produced by the device.

VII.1.3. Chinese Teletypewriter

At the time of writing this report, the project has acquired the Chinese teletypewriter system developed by Mr. Chung-chin Kao. We are at the exploratory stage of interfacing it with our system. It seems that this device, which offers a hard-copy of the input for proof reading, will help to improve our text coding procedures.

VII.2. The Output System

The SAS is able to produce two alternative hard-copy output after a particular sentence has been subjected to analysis and interlingual conversion.

VII.2.1. Printed Output

The structural trees which result from the analysis were printed in indentation format in order to identify the different substructures of the tree. However, a long string may easily result in a tree which can have 35 levels or more. Moreover, the complex references to partially ambiguous substructures within these levels further complicated the task of our staff in coming to a better understanding of the output without graphic aid.

During the first part of our contract period, graphical representation of these printouts was by the laborious means of hand-plotted trees using data directly from these printouts.

VII.2.2. Calcomp Plotted Output

One of the major achievements during the latter part of our contract period was the successful implementing of a set of programs to plot these trees on the Calcomp plotter. The system has the capability of plotting both 12" and 30" wide plots, depending on the number of levels.

In addition to writing the telecode string on the plot itself, our graphic system also plots the corresponding Chinese characters. This has proved to be a most valuable aid in identifying the sentences under examination.

The plotting of the structural tree identifies each node by its grammar code. Each terminal node also has associated with it the terminal grammar code, the telecodes, romanization and English gloss. Each branch is also identified by the character positions in the sentence which is dominated by that node.

All possible ambiguous structures also appear as part of the whole plotted tree and are identified as alternative branchings by using dashed lines rather than solid lines.

These plotted outputs show at a glance whether a particular sentence has received the correct analysis or not. The ambiguous structures on the plot are extremely useful in aiding the linguist to prune illegitimate ambiguities from the rules of the grammar as well as setting up more adequate rules for the sentence.

The interlingual work also makes use of these plotted trees as a diagnostic aid, since the relevant nodes involving interlingual rules are also indicated in the plots of both the Chinese and English trees for each sentence.

At the close of the contractual period, the electrostatic plotter's capabilities were also explored. The high speed which this type of plotter is capable in producing a page of output offers advantages for high volume output of Chinese character texts, for example in concordance work. It is less adaptable for output of plotted trees.

VIII. Structure of Syntax Analysis System (SAS)

VIII.1. General Strategy

The processing of Chinese text by the Syntax Analysis System can best be described in the following logical sequences:

VIII.1.1. Inputting of Text

The following modes of physical input are available:

(a) punched cards

(b) Chicoder conversion (see discussion on use of telecode coding)

VII.1.2. Initialization Phases

(1) Subdictionary selection. A subset of the CHIDIC entries is selected which is relevant to the field of the text being processed. Since CHIDIC now has entries for biochemistry, physics and other general vocabulary, it is more economical and efficient in terms of programming to use a subset of the dictionary only.

(2) Dictionary update. After selection of entries from CHIDIC, this subdictionary is updated with new entries which occur in the text and other entries which have been compiled during the period preceding the run.

(3) Rule update. The grammar rules must be updated to reflect changes in the rules since the last text was processed, as well as changes following grammatical studies on particular areas of grammar which result in the addition and/or deletion of grammar rules.

(4) Rule adaptation. After a new version of the grammar rules has been updated, this program adapts the rules into binary format. This is necessary since some grammar rules have as much as quinary (or 5-ary) branching.

(5) Dictionary adaptation. The format of the selected subdictionary must also be adapted to the correct internal format.

VIII.1.3. Main Processing Phase

(1) Pre-edit. This routine processes the coded text to adapt the strings for the look-up and parsing stages. During this stage, punctuation marks, paragraphs, special symbols, etc., are flagged for special treatment.

(2) Look-up. Each parsing unit (string of telecodes) is scanned left to right and right to left using the longest match technique to look-up entries from the subdictionary. All information associated with the successfully found entry is retained. These include the grammar codes, telecodes and English gloss.

(3) *Parsing*. This may be considered the most vital phase of processing. Each string is processed for all possible legitimate structures, using the adapted rules. Results of possible structures are entered into a constitute table, which can be printed for visual inspection. Information regarding whether a string is parsed or not parsed to a highest node which spans the whole string is also printed.

(4) *Interlingual (UPROOT)*. As soon as structural information for the successfully parsed string is available, the interlingual routines make use of this to adapt the Chinese structural tree to its equivalent English structure. The most successfully implemented routines are those which permute a subtree around its node.

VIII.1.4. Output

Two main types of output are available:

(1) *Printer output*. The string in telecodes, the results of look-up, the results of parsing (constitute tables), the Chinese and English structural trees (UPROOT) plus diagnostic information are all printed on the printer output.

(2) *Plotted output*. Two stages are required:

(a) The SAS computer program completely specifies the form of the plotted output, line by line and letter by letter, and produces an intermediate file with this information.

(b) The Calcomp Program of GDS reads the intermediate file and translates it into the hardware conventions for the Calcomp plotter. The resulting information is submitted to the plotter to be drawn on either 12 inch or 30 inch paper. (Because the only input device attached to our local plotter is a card reader, communication between the GDS program and the plotter is through a deck of punched cards.

IX. Overview of the Logic of The Present Syntax Analysis System

The present Syntax Analysis System has three phases: preparation, text processing, and run termination. It is implemented with 54 routines in 9 overlays.

The ten system flowcharts (Appendix I) display the calling relationships between the routines. On these flowcharts the nonparenthesized name in each block is the entry point or overlay called. If the name of the routine is different from the name called, it is included in parenthesis. In the upper right-hand corner of each block is a pair of numbers designating the overlay containing the routine. In the lower right-hand corner is either an F or a C depending on whether the routine is in FORTRAN or in COMPASS, the assembly language for CDC 6000 machines.

IX.1. Preparation Phase

The preparation phase is represented by the first three system flowcharts. Program MANAGER begins by reading the run parameters from a lead card. The first two lead card parameters determines the mode of rule and dictionary preparation, respectively (see MANAGER documentation).

Flowchart 1 represents the preparation phase of a normal run without adaptation of rules or dictionary.

If the preadapted rules are to be read into Extended Core Storage (ECS) from tape, subroutine RR is called.

If freshly updated rules are to be used instead, then the call to overlay 10 on flowchart 2 replaces the call to subroutine RR. In this case AR, the main program of overlay 10, adapts the newly updated rules (see AR documentation) calling, in some cases several times, the various routines under it on flowchart 2. Whether new rules are adapted or not, if a printout of the adapted rules is requested, subroutine PR is called.

The dictionary preparation is similar, If the second lead card parameter specifies that a preadapted subdictionary is to be read into ECS then subroutine RD is called as on flowchart 1. If on the other hand a newly updated subdictionary is to be adapted then the call to overlay 20 on flowchart 3 replaces the call to RD on flowchart 1. Again the printing is specified separately and may be done whether or not a new dictionary has been adapted. The printing of the dictionary is done by program PD of overlay 30, represented on flowchart 3.

Then back on flowchart 1 APILT is called to Adapt and Print the Interlingual Transformations. This completes the Preparation phase.

IX.2. Text Processing Phase

The high level text processing calls are represented in flowchart 4 with their subcalls on flowcharts 5 through 9. MANAGER begins the text processing phase by calling overlay 40 (see flowchart 5 for the subroutine calls under overlay 40). PREEDIT, the main program of overlay 40, prints a sentence header using information including time and date information obtained from the operating system through a call to entry point WHEN of subroutine RUNID. PREEDIT then determines how much of the text is to be regarded as the first sentence or parsable unit. For convenience we will refer to parsable units as sentences from now on even though occasionally they happen to be merely headings or miscellaneous fragments included in the text. The first sentence is then printed and PREEDIT calls LOOKUP which does a longest-match lookup of the text using the adapted dictionary in ECS. LOOKUP thereby builds a table of terminals in which the grammar code is represented by its index in the grammar code table. Finding this index is the only use LOOKUP makes of the utility binary search routine DSRCH. When the terminal table is complete, LOOKUP returns control to PREEDIT which in turn returns control to MANAGER, which then calls overlay 50.

SYNTAX, the main program of overlay 50, then calls overlay 51 whose calls are represented on flowchart 6. RTTOLFT, the main program of overlay 51, processes the sentence from right to left. For each sentence position it constructs terminal constitutes for all the relevant entries in the terminal table that begin with that sentence position, and then calls PARSE, which constructs all of the non-terminal constitutes which begin at that sentence position. (For the parsing logic, please see the documentation on PARSE.) When RTTOLFT is finished with the sentence, it returns control to SYNTAX in overlay 50 which calls PRNT (see flowchart 7) which will print either a table of treetops found in parsing or a table of breaks showing where parsing failed. Depending on a table of parameters read from the lead card by MANAGER, PRNT may also print the entire constitute table or, through repeated calls to overlay 53, print all of the maximal partial trees for the sentence if it was not parsed.

When PRNT is finished it returns control to SYNTAX. If the sentence was parsed, SYNTAX then calls overlay 53 to extract the Chinese tree from the constitute table. Depending on lead card parameters UPROOT, the main program of overlay 53 will either print the tree or prepare tables from which it may be plotted, or both (see flowchart 8 for calls from UPROOT).

UPROOT returns control to SYNTAX, which then calls overlay 54 if plotting was requested on the lead card (see flowchart 9 for calls from PLOTREE, the main program of overlay 54). PLOTREE makes numerous calls to the various entry points of TOMFUNG, the interface between our Syntax Analysis System and the Computer Center's proprietary Graphics Display System (GDS). In making these calls PLOTREE uses as parameters the data stored in tables by UPROOT.

The Graphics Display System writes an output tape which will be read later by the GDS routine CALCOMP.

When PLOTREE finishes making plotting calls, it returns control to SYNTAX in overlay 50. If English trees also were to be printed or plotted, SYNTAX then calls overlay 53 again, and this time UPROOT calls TRNSFRM once for each node. This causes all interlingual transformations to be applied and the resulting structure is the English tree. If the English tree was to be plotted, then overlay 54 is called again, this time with plotting tables for a transformed tree.

SYNTAX then returns control to MANAGER clear up in overlay 00. MANAGER then calls overlay 40 again, PREEDIT picks out the next sentence and the entire text processing cycle is repeated until either the text is exhausted or the internal time limit (the last lead card parameter) has expired. For uniformity, if the text is exhausted, the internal time limit is reduced, forcing its expiration.

IX.3. Run Termination

The small routine TIME in overlay 00 is called frequently throughout the system. Primarily it prints the elapsed time for the various subprocesses, but each time it is called it also checks the overall elapsed time for the entire run against the internal time limit. When the internal time limit has been exceeded the run termination phase has begun and the calls on flowchart 10 are executed.

It is merely to provide the Graphics Display System with the opportunity to do its end-of-run processing and file closing that we execute the chain of calls from time, through overlay 50, to overlay 54 and ENDPLOT.

When control has been passed back to TIME in overlay 00 it then calls overlay 40 in which PREEDIT prints the table of ambiguities encountered during the run, and the table of sentences parsed.

When the Syntax Analysis System has stopped, the GDS program CALCOMP rewinds the tape prepared by the Graphics Display System and prepares from it the deck of binary cards which will drive the hardware CALCOMP plotter.

IX.4. Highlights of Software-Hardware Interface

As has been stated previously, the difficulties arising from the non-alphabetic script of Chinese affect directly the efficiency of input/output. Efficiency in I/O of Chinese characters has plagued Chinese MT since its inception. Technological advances in recent years have helped to make the task somewhat less arduous. One input problem in particular is the several levels of processing required in order to have an adequate corpus of machine readable text. That aspect of coding has already been discussed. As regards the output, it is even more essential that a scheme has to be devised to enable the human reader to study the results of the machine analysis.

During the first part of the contractual period, the analyses were entirely produced on the line printer. Therefore only the coded Chinese characters were available (plus romanization). Furthermore, the analyzed tree structures could not be plotted by machine; this compounded the difficulties of post machine reanalysis, since it meant that each tree had to be manually drawn from the output data. Considering that for each run about 100 sentences are processed, the considerable time lag between the results of a run and the reanalysis of the results was rather obvious.

The solution was to obtain graphic output as directly as possible via machine. The choice of hardware was partially decided upon by the readily available Calcomp Plotter. Its system software had already been developed at the Berkeley Computer Center, which enable us to use the output from our SAS with a minimum of reprogramming. A drum plotter such as the Calcomp is the most advantageous choice for producing tree diagrams, because the only information which must be provided to it are the endpoints of the lines to be drawn. This means that we are able to specify lines in the logical sequence in which they occur during analysis, without having to associate all information which will eventually end up in any particular sector of the plot. The drum plotter translates the lines into uniform steps along the X and Y axes, re-entering each sector as often as necessary. If such a procedure were not possible, a great deal more computation and internal storage would be required, since our plots are 30 inches wide and typically between 8 feet and 25 feet long.

A comparison may be made to some recent developments in plotting hardware, such as the electrostatic plotter. This type of plotter requires that a page be output at a time. The total information for that page of output must thus be completely stored before output, precisely what is undesirable for plotting syntax trees. Furthermore, the page size is quite limited on current electrostatic plotters, and a large tree diagram would have to be manually assembled from the small pieces produced. On the other hand, this fixed grid method of plotting is very advantageous for large scale output of text. For a page of text to be output on such a machine, the advance storing of a whole page of characters is quite feasible and efficient since the information for a page is logically developed in sequential order. The speed factor is particularly attractive. Initial comparisons indicate that it takes as long for the Calcomp to plot one Chinese character as it takes to output a whole page of characters on an electrostatic plotter. The latter's application to concordance work immediately suggests itself. The electrostatic plotter is rapidly passing through its technical development stages and would appear to be quite promising for such applications. (The plotting unit itself is also quite portable.)

We have stressed the programming aspect of our plotting system because it is quite software oriented. The plotting of Chinese characters at present makes use of the character vectors coded by Professor Kuno of Harvard University. One advantage of our software lies in the fact that it is now possible to plot not only labeled trees, telecodes and romanizations of Chinese characters, but also to display the Chinese characters themselves on the plot.

The ability to display the results of our analysis for each sentence in terms of its analyzed structure, and the associating of Chinese characters with this structure is a solid contribution to increasing the efficiency of research on the Project. This is particularly so, in view of the planned concordances of Chinese texts that are expected to be produced as an aid to our present research.

For detailed documentation of the Syntax Analysis System please see Appendix II.

X. Auxiliary Diagnostic Processes

These routines are not part of the SAS package. They perform services which are essential to the maintenance of SAS.

X.1. Updating Data Base

The whole of CHIDIC is periodically updated as more entries are compiled. The same routine is used for updating both CHIDIC and the subdictionary.

X.2. Concordance

(1) Rule Concordance. This routine will concord the whole set of grammar rules, listing the occurrence of each grammar code type and its tokens in each rule.

(2) A program has been written which will concord texts based on certain syntactic boundaries rather than on a strict length per line basis. For example, for a particular token which occurs within a clause, the whole clause will be printed unless this clause exceeds some particular length limit. Thus if the token is the first word in the sentence, then no output is printed for that portion of the preceding sentence to the left of the token. Rather, the position of the concorded item is shifted left or right on the page to accommodate the largest possible syntactic unit within that sentence. This is a departure from the general output format, such as that of the KWIC index. The

emphasis here is to preserve the whole relevant syntactic unit in which the token occurs. For syntactic analysis by the linguist this kind of information is most helpful.

X.3. Random Generation

This program will randomly generate sentences based on a restricted set of our grammar rules.[1] Preliminary documentation was completed (See Appendix V). It is planned to further restrict the direction of generation in order to use this problem as a diagnostic aid in improving the present grammar.

X.4. Character Plotting

A program package which makes use of the CDC 6400 Graphic Display System package plots Chinese characters using the Calcomp plotter. The coded information for each character has been made available to the Project by Dr. Susumu Kuno of Harvard University.

X.5. Break Tables, Constitute Tables

Built into the Parser routine are options for sentence analysis diagnostics. The Break Tables provide a list of the "breaks" or gaps in the grammar rules, indicating no higher constitutes could be formed. The Constitute Tables provide the full set of possible constitutes into which any particular string was analyzed. In combination, these two diagnostics provide for very effective post analysis of every string submitted for machine analysis.

---

[1] Random generation of sentences based on a parsing grammar should be differentiated from random generation based on a generative grammar. The aim of a parsing grammar is to provide correct structural descriptions of input sentences, which are well formed. A generative grammar aims at producing only well-formed sentences. For contrast, see T'sou, Benjamin K. 1963. "Chinese Grammar I: Random Generation of Adjectival Modifier Constructions" (Internal Report), Research Laboratory of Applied Electronics, Mechanical Translation Group, Masschusetts Institute of Technology.

XI. Analysis of Processed Text

During the contractual period, scientific texts in both nuclear physics and biochemistry were studied, analyzed, coded and subjected to analysis by the Syntactic Analysis System. As of the end of this period, the Project has available 106 pages of coded, machine readable text, consisting of various articles from the journals Yuantzu Neng ('Atomic Energy') and Acta Biochimica Sinica.[1]

Before a text was actually submited for mechanical analysis, it was normally preceded by preliminary processing which consisted of updating the grammar rules and lexical entries as a result of varying levels of analysis on the text. Texts which did not receive such analysis were also submitted for processing to act as control in assessing the ability of the system.

XI.1. Run Statistics

The accompanying tables represent results of runs made at the conclusion of the contract period. The tables indicate the percentage of parsing success with respect to sentence length.

Tables 1A and 1B show the results of a second run on a

---

[1] The corpus of new text selected for processing under this Contract was drawn exclusively from Yuantzu Neng. Some selections from Acta Biochimica Sinica had been retained for the purpose of controlling improvements in grammatical analysis.

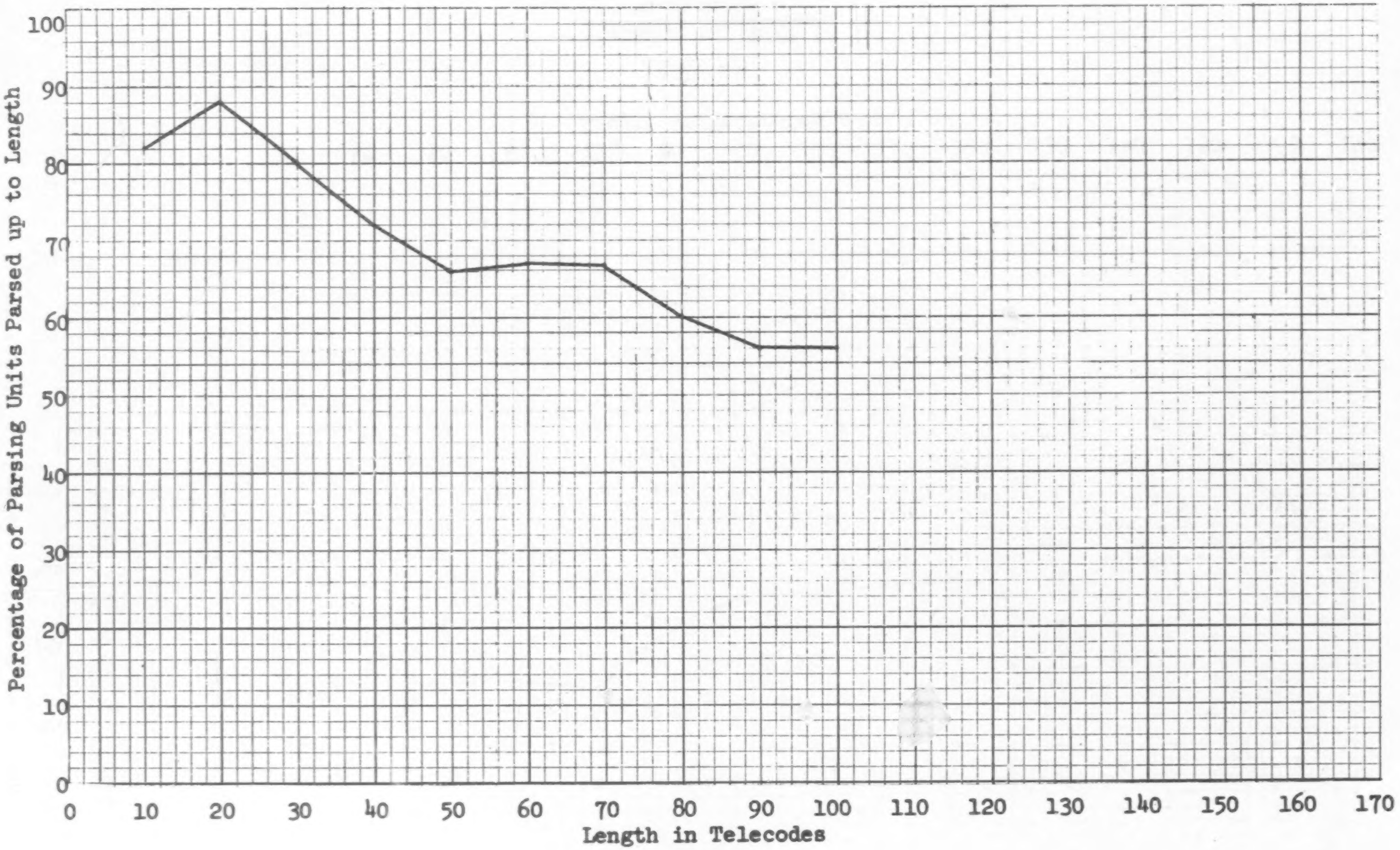


TABLE 1A

BIOCHEMISTRY TEXT I
All Parsing Units

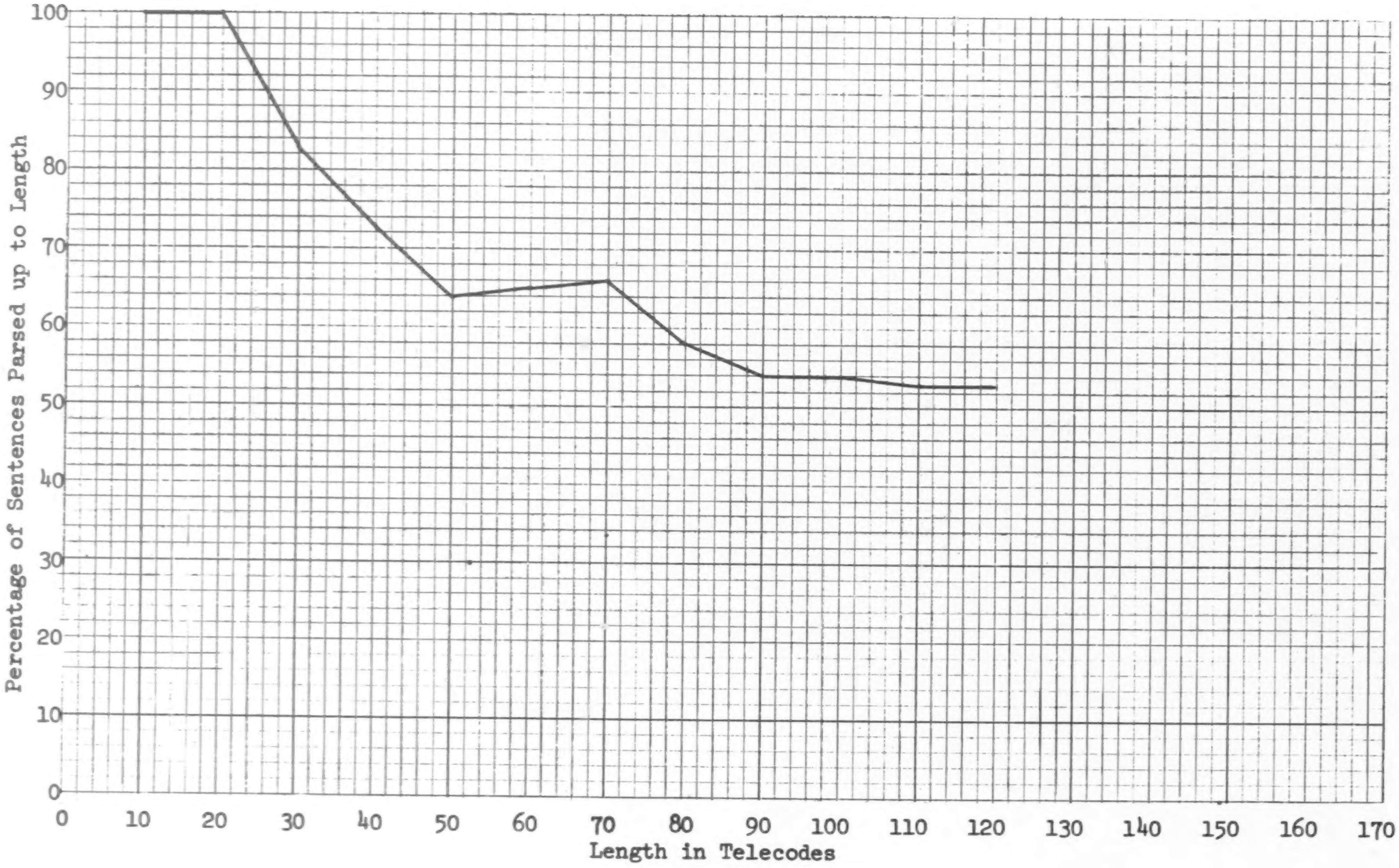


TABLE 1B

BIOCHEMISTRY TEXT I
Sentences

biochemistry text after detailed re-analysis of results of the first run, which was made during the mid-period of the contract. (The results of this initial analysis was discussed in the quarterly contract status report for the period June-September 1969.)[2] The results of the second run showed significant improvement (64%) over that of the first run (44%) for sentences and for all parsing units up to 50 telecodes in length.

Tables 3A and 3B represents the combined results of 2 runs of biochemistry text which were not subjected to the sort of preliminary processing and re-analysis that was made for the text of Tables 1. As a result, the percentage of units parsed would not be expected to be as high as that for the previous text. However, in spite of this the percentage stands around the 30%-40% area, indicating the ability of the SAS to maintain reasonable consistency in parsing units of 50 telecodes or less. It should be noted that during the interim report it was only after the first text had been subjected to a minimal amount of pre-editing that a 40% success was obtained for sentences up to 50 telecodes in length. The much higher percentage obtained for this same text for the second run represented approximately six months of effort between the two runs.

Tables 4A and 4B show the results of a sample of physics text which was run after 3 to 4 months of preliminary processing and analysis. The percentages according to the 50 telecode length

---

[2] The tabulated results are included here as Table 2.

TABLE 2

	SENTENCES ONLY						ALL PARSABLE UNITS					
	TOTAL NUMBER	LOOKUP COMPLETE	PARSED	% OF T,N. L.C.	% OFT.N. PARSED	% OF L.C. PARSED	TOTAL NUMBER	LOOKUP COMPLETE	PARSED	% OF T.N. L.C.	% OF T.N. PARSED	% OF L.C. PARSED
FIRST RUN:												
0-50	52	51	23	98	44	45	75	74	36	99	48	49
50-100	29	26	5	90	17	19	29	26	5	90	17	19
100-150	3	3	0	100	0	0	3	3	0	100	0	0
0-100	81	77	28	95	35	36	104	100	41	96	39	41
0-150	84	80	28	95	33	35	107	103	41	96	38	40
SECOND RUN:												
0-50	10	10	3	100	30	30	22	22	6	100	27	27
50-100	8	6	1	75	12	17	8	6	1	75	12	17
100-150	4	3	0	75	0	0	4	3	0	75	0	0
0-100	18	16	4	89	22	18	30	28	7	93	23	25
0-150	22	19	4	86	21	18	34	31	7	91	22	23
COMBINED:												
0-50	62	61	26	99	42	43	97	96	42	99	43	44
50-100	37	32	6	87	16	19	37	32	6	87	16	19
100-150	7	6	0	86	0	0	7	6	0	86	0	0
0-100	99	93	32	94	32	34	134	128	48	96	36	38
0-150	106	99	32	93	29	32	141	134	48	96	34	36

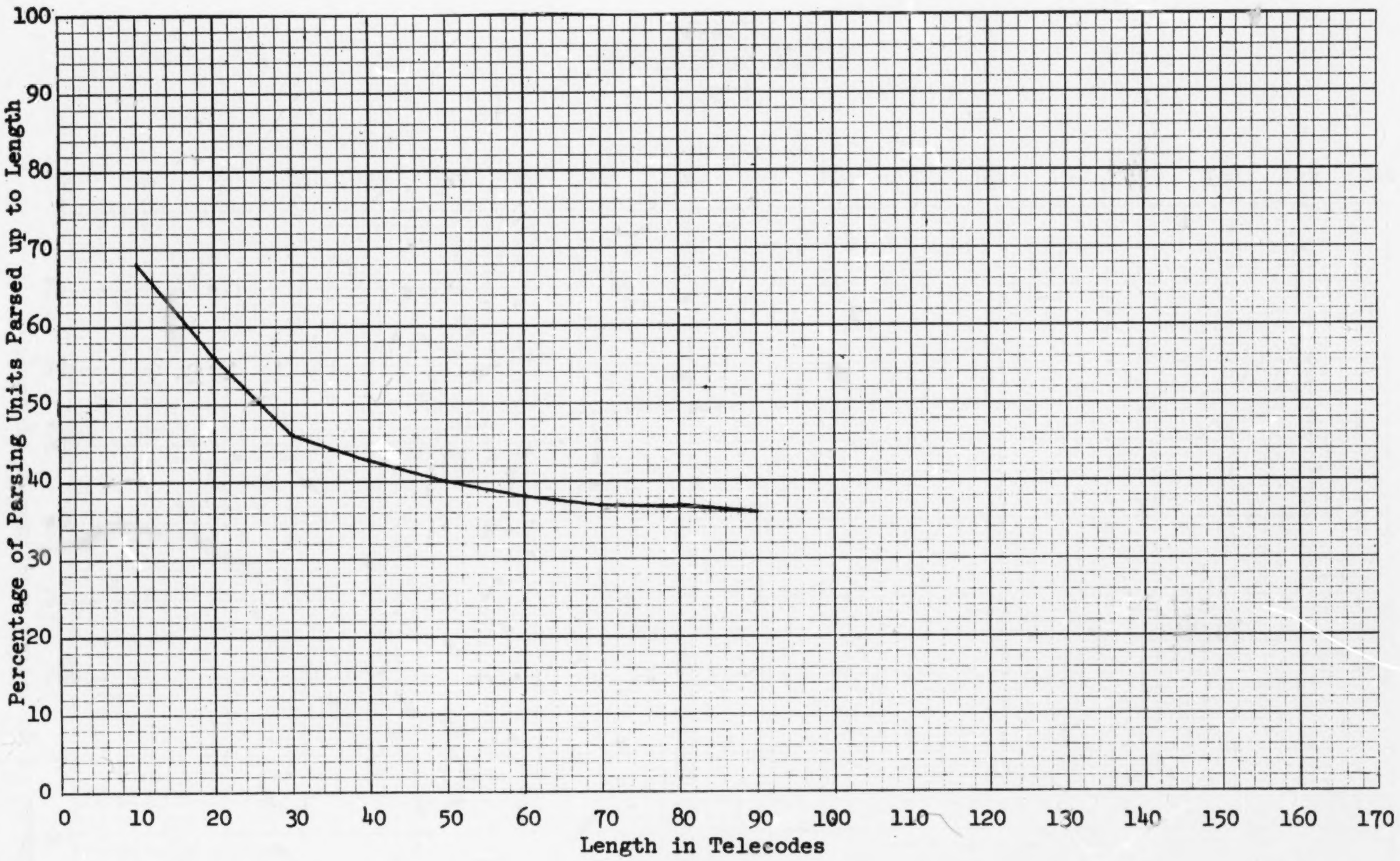


TABLE 3A

BIOCHEMISTRY TEXT II
All Parsing Units

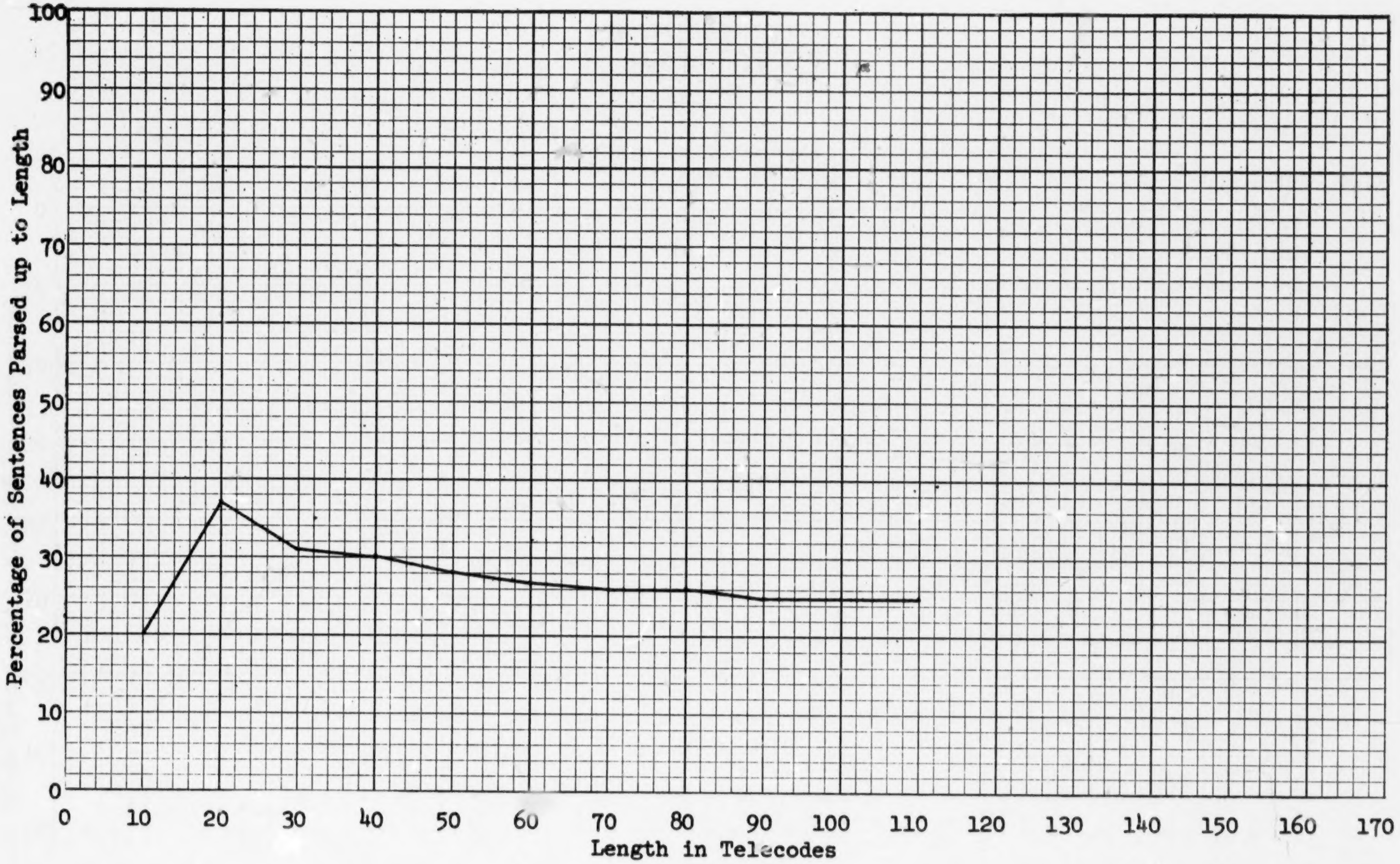


TABLE 3B

BIOCHEMISTRY TEXT II
Sentences

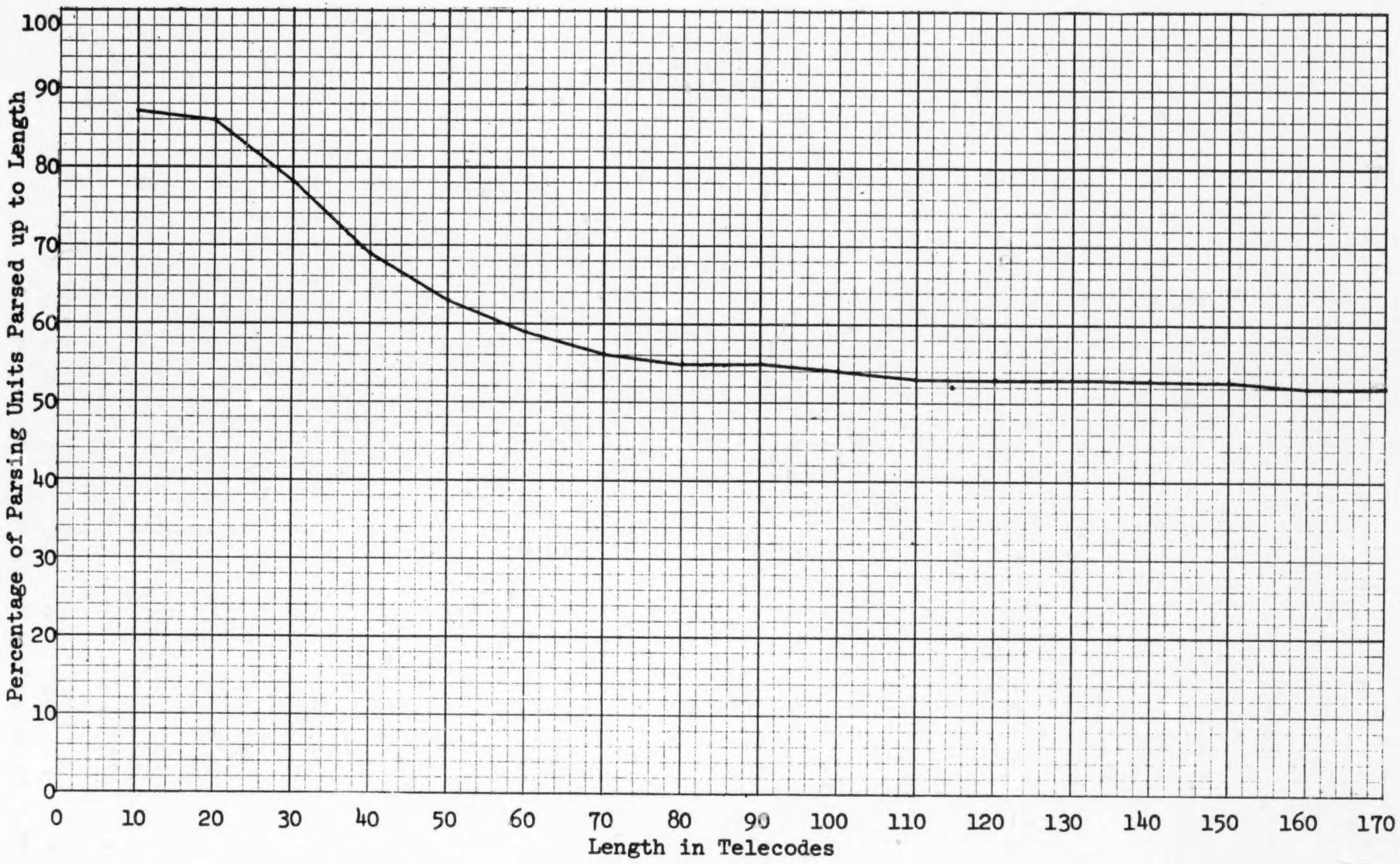


TABLE 4A

PHYSICS TEXT I
All Parsing Units

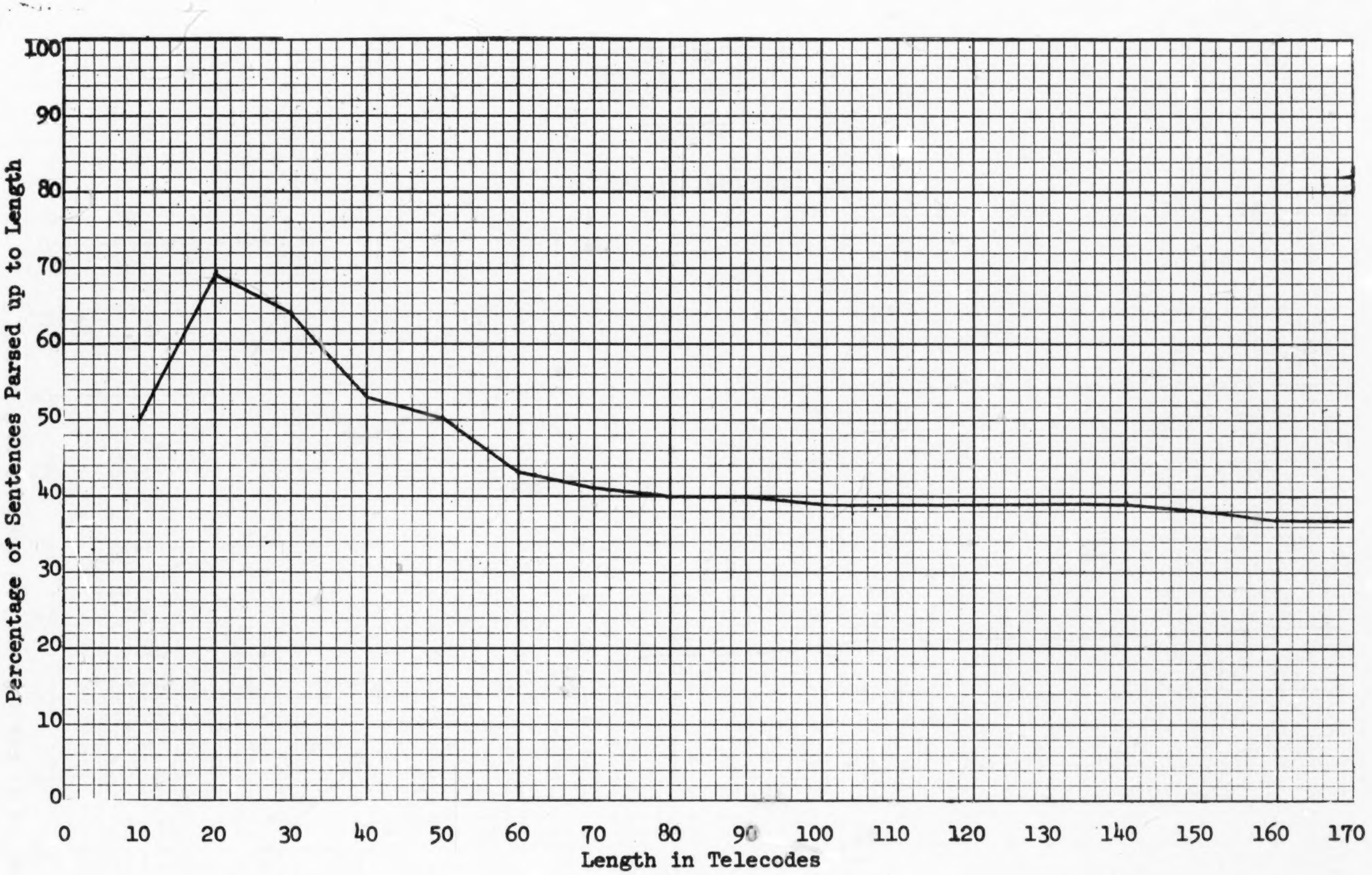


TABLE 4B
PHYSICS TEXT I
Sentences

criterion were slightly lower than those for the pre-analysed biochemistry text, but were significantly higher than the virgin text.

Several factors must be taken into careful consideration in evaluating the results. Particularly the percentage increase should not be taken as a linear function of efforts in analysis. The main factors which affect the results are discussed below:

(1) Small sample. The texts represent an extremely small sample. Each sample has less than 200 total parsing units. The results are therefore only specific, not representative.

(2) Subject matter. Two different subject areas -- biochemistry and nuclear physics -- were treated. Biochemistry had been the area of concentration in work prior to the present contract. Therefore, the dictionary, as well as the grammar rules directly reflect this prior concentration. The special area of concentration has been shifted to that of nuclear physics during the present contract. Efforts were therefore directed toward acquisition of nuclear physics lexical entries for the dictionary. At present physics entries form less than one quarter of the total dictionary. The results obtained from running an unedited physics text would not have been meaningful. However, from the comparable results of pre-edited biochemistry and physics texts above, it does seem that work on the biochemistry and physics text I has carried over some of the parsing ability of the grammar to the physics text.

(3) Sentence length as a criterion. In the above tables, we have used the 50 telecode length as a key for comparison. This may seem like an arbitrary choice; and, in a sense, it is so since at this stage of research in Chinese sentences, there is not yet sufficient machine readable text to obtain any good statistics regarding sentence length characteristics.[3] It is not sufficient to evaluate the system solely in terms of number of telecodes although this does give a certain quantitative criterion. However, from our actual experience with analysis of sentences, it is in the range around 50 telecodes that the parsing ability of the SAS begins to decrease more rapidly. For sentences of about 20 telecodes in length, parsing success is in the 90% range. But for sentences of length greater than 20 telecodes, it was found that quite a number of these are instances of conjoined sentences (either coordinate or subordinate); or they are sentences with several levels of embeddings. At present our grammar lacks refinement in handling such complex sentences. Since our grammar can already process satisfactorily units of up to 20 telecodes, we expect that if procedures for isolating simple sentence structures within these complex sentences could be devised the number of parsed units should increase for sentences with a greater number of telecodes, provided these are genuine conjoined structures linking simple sentences together.

---

[3] Average sentence length is about 34 for both biochemistry and physics texts that had been processed so far.

(4) Limiting factor of CF grammar. As was already mentioned in preceding chapters, the CF framework limits the ability of the system as sentences become more complex. It is our task to refine and improve these CF rules, but it should not be expected that the addition of more rules will better the ability of the grammar to parse more and more complex sentences. Within a connected text, there are many syntactic and semantic points of cross-reference, which are characteristic of discourse. These add to the complexity of the text as a whole. The CF framework could handle such cross-reference only very inadequately.

(5) Variability in style. As compared with English, style variability in written Chinese is more complex. From our study, we can single out two areas

> (a) Synchronic. By this we refer to the variability that exists from one text to the next due to differences in authorship and subject matter. Anyone who has dealt with texts is familiar with the complexity of this type of style variation. It is also within the scope of present theoretical syntactic research to deal with such problems.
>
> (b) Diachronic. A mixture of modern literary Chinese and classical Chinese exists in all the texts we have studied while analysis of texts in the modern written European languages do not

present such a problem.[4] It means the parsing grammar must be able to deal with "standard" modern written Chinese but also have to cope with different varieties of written classical Chinese. This taxes the ability of the Berkeley grammar to its limit.[5]

In summary then, the run statistics presented in the above tables are indicative of texts which underwent vigorous analysis. The results of such analysis have improved the ability of the system as a whole. However, in processing raw text, the range of variability will be great on account of the factors just mentioned. Realistically, the parsing percentage can be lower from those presented. The factors are such that the above-mentioned increases in parsing percentage cannot be expected to indicate linear improvement for the system as a whole.

XI.2. Plotted Chinese Trees and Interlingual Trees

Appendix III presents a set of 8 plotted tree structures representing the analysis of 4 sentences. Each sentence is first analysed in its Chinese linear string order. The following plot shows the resulting structure after interlingual procedures have applied. (These plots were abstracted from our 30-inch Calcomp plots to facilitate discussion.)

---

[4] Comparable cases exist for other written traditions. Modern literary Arabic for example uses a mixture of classical and modern Arabic. Modern literary Thai also makes use of colloquial Thai and classical Sanskrit elements.

[5] This has necessitated the project to devote some efforts to classical Chinese for there is a notable scarcity of relevant literature in this field.

XI.2.1. Chinese Tree

The dashed lines indicate alternative analysis for segments of a string. For example, in Sentence 1, the 3 dashed lines radiating from the node VI3 indicate there are 3 alternative analyses of the string which lead to the node VI3. Nodes which are not expanded downward to terminals indicate that the subtree of this node is the root which already exists in the plot. (In the actual plots cross-references to these nodes are noted automatically on the plots.) For example, the nodes NP*R all have the same structure as the left most NP*R.

On the plots, the alternatives are not linearly ordered, in the sense that the one on the left is to be considered the best analysis, the second left, less so, and so on. The alternatives all have the same validity as far as the grammar is concerned. It is only in the post-mortum analysis stage that the human analyst can decide which, if any, of the alternatives are acceptable in terms of the context of a particular text.

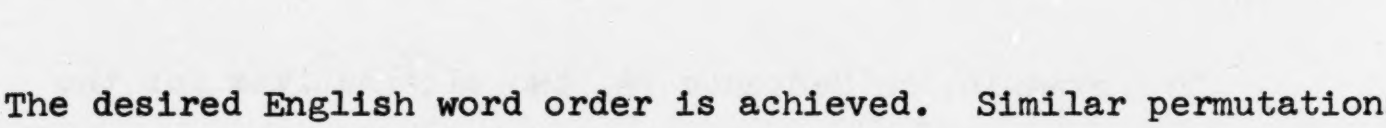
For example, in Sentence 1A, two alternatives for the entry hou 後 appear. The first one has grammar code NL (place noun) and glossed 'rear'. The second has grammar code 1V-NT (a time noun suffix), glossed 'after'. On examination, it can be decided that the second alternative has more validity and the human analyst will continue to examine the rest of the structure connected with this second analysis.

XI.2.2. Interlingual Tree

The B sentences show the results of interlingual transformation on the A sentences. We have already discussed the interlingual processes in Chapter VI (Interlingual Translation). Continuing with our example in Sentence 1A, the entry hou with grammar code 1V-NT participates in an interlingual permutation due to the rule

NT*R → IND + 1V-NT .

The resulting structure (in Sentence 1B) shows that the string under NT*R have been reordered:



The desired English word order is achieved. Similar permutations are performed whenever a node with *R occurs.

The ability to plot the original structure of the Chinese sentence and the resulting interlingual structure has provided a visual aid which greatly increased the efficiency of post-mortum analysis.

XI.2.3. Ambiguities

While the output of the system provides numerous alternatives, including some spurious ones, it should be pointed out that ambiguities per se do not invalidate the parsing ability of the system. There are occasions when the grammar discovered legitimate ambiguities that the linguist was not yet aware of. It points up the exhaustive nature of analysis by machine and the advantages of subjecting human analysis to such an impartial tool as the computer. However, examination of the plots also gives us a very good indication of where ambiguities might be pruned.

Referring to Sentence 1 again, there are actually 7 alternative analyses for the sentence. Two of the alternatives of the AV2 node are very similar and call for further refinement of the grammar rules with possible reduction of ambiguities. It is expected that at a later stage of our research, when English string extraction is implemented, the number of string representations would be far less than the actual ambiguities as presently represented on the plots.

XI.3. Evaluation of English Output

During the present contractual period emphasis was necessarily placed on obtaining correct analysis of the input Chinese string which would eventually result in as few ambiguous English translations as possible. Previous chapters have already discussed the role of the interlingual transfer rules, the restrictions in the dictionary to the field of nuclear physics, and the lexical disambiguations which will be required in both the pre-parsing and possibly post-parsing stages.

As evidence by the output represented by the English structural trees, two interacting general areas regarding refinement to the English output will have to demand a great deal of future effort.

(1) *Reduction of invalid and irrelevant ambiguities* via analysis of the structure of the Chinese input string. This is the direction of development in our present effort, as was already mentioned in the previous paragraph and in the foregoing chapters of this report.

(2) *The reduction of English string paraphrases*. The legitimately produced ambiguities of (1) above may also incorporate within themselves several paraphrases for each ambiguiously produced string. Paraphrases are, of course, not ambiguities. (For an example, see the sentences (1a),(1b), (1c) under section on Lexical Disambiguation.) However, they do represent a proliferation

on the total number of English strings that are available for the corresponding Chinese string and the English synthesis component of the system must take account of this fact.

XII. Summary and Conclusions

During the two-year contractual period (September 1968 - August 1970), the Project on Linguistic Analysis has expanded the size of the dictionary (CHIDIC) to 57,000 lexical entries. Although it is possible to access all these entries directly, it was more efficient to use a subdictionary of about 3,000 entries obtained from CHIDIC during each run of text. Another advantage was the ease with which any changes may be made in the subdictionary without the necessity of overhauling the whole of CHIDIC. The later task is undertaken during the periodic updating of the whole dictionary.

The main tasks have been the refinement and expansion of the grammar and the improvement of the programming routines in the Syntax Analysis System. Two major runs of continuous text were made at the end of the first year and also at the end of the second year. It was found that SAS can recognize and parse satisfactorily strings up to 25 characters long with comparatively low degrees of ambiguity. But that ambiguity increases with increase in length. As long as the grammar rules are in the form of context-free phrase structure grammars, many spurious ambiguities will arise. Since this type of grammar is still the most efficient to implement, the resolution of ambiguities must also be supplemented by other procedures besides pruning redundancies within the grammar rules themselves.

Aside from the dictionary and the grammar, within the present system there are several areas where procedures may be specified in order to reduce the amount of ambiguity.

(1) In the pre-editing program. This is conceptually the simpler procedure since it will require a fixed table from which particular lexical ambiguities may be resolved. Discontinuous constitutes which occur frequently may be placed in such a table and associated with particular addresses which specify certain editing procedures. Certain procedures may in fact be equivalent to particular "transformations" on the string in order to edit them into normalized form for parsing.

(2) Features in the parsing program. In the chapter on syntax we have discussed problems of ambiguity involving active vs. passive, compounding, nominalization, deverbal nouns, etc., which require more detailed information regarding their feature selection properties with respect to one another. It is most important that during the parsing stage these features, which are supplied to the parser by the dictionary, be highly compatible in their mutual selectional properties. This basically extends the capabilities of the present system, which already incorporates to a lesser extent certain selectional features within the grammar rules and the grammar codes assigned to each lexical entry in the dictionary.

The task of zeroing in on the most acceptable analysis after the sentence has been analyzed was aided tremendously by having each sentence so analyzed graphically plotted to show their different tree structures. Examining such plots in graphic representation has enabled us to determine particular nodes in these trees where interlingual rules may profitably be created and applied in the Chinese and English trees.

During the present phase of development we have achieved significant results in analysing Chinese sentences. While we have also implemented procedures for interlingual transfer the results have not been as concrete since the state of knowledge regarding Chinese-English contrastive studies and their application in the area of computational linguistics is still in its infancy.

BIBLIOGRAPHY

Bar-Hillel, Yehoshua. 1970. "Position Paper on MT in 1970", Texas Symposium on Machine Translation.

Bever, Thomas G. and Peter S. Rosenbaum. 1970. "Some Lexical Structures and Their Empirical Validity", in Readings in English Transformational Grammar. (Eds. Roderick A. Jacobs and Peter S. Rosenbaum) Boston, Mass.

Binnick, Robert. 1967. "The Lexicon in a Derivational Semantic Theory of Transformational Grammar", Chicago Journal of Linguistics, Vol I. University Microfilm, Ann Arbor.

___________. 1968. "On the Nature of the 'Lexical Item'", Papers from the Fifth Regional Meeting, Chicago Linguistic Society. (Eds. B.J. Daden, C.-J. N. Bailey, and A. Davison) University of Chicago, Department of Linguistics.

Boas, Franz. 1970. Introduction to the Handbook of American Indian Languages. University of Nebraska Press, Lincoln.

Catford, J. C. 1969. A Linguistic Theory of Translation. Oxford University Press, London.

Chafe, Wallace L. 1970. Meaning and the Structure of Language. University of Chicago Press, Chicago, Illinois.

Chao, Yuen Ren. 1968. Language and Symbolic System. Cambridge University Press.

___________. 1969. "On Translation". Taped lecture given at UC Berkeley. Berkeley, California.

Chapin, Paul. 1967. "The Syntax of Word-derivation in English", M.I.T. PhD. dissertation. Mitre Corporation, Bedford, Mass.

Chomsky, Noam. 1956. "Three Models for the Description of Language", IRE Transaction On Information Theory Vol. IT-2, Proceedings on the Symposium on Information Theory. September.

Chomsky, Noam. 1957. *Syntactic Structures*. The Hague, Mouton.

___________. 1965. *Aspects of the Theory of Syntax*. The M.I.T. Press, Cambridge, Mass.

___________. 1966. "Topics in the Theory of Generative Grammar", in *Current Trends in Linguistics*, Vol. 3: *Theoretical Foundations*. (Ed. Th. A. Sebeok) The Hague, 1-60.

___________. 1970. "Remarks on Nominalization", in *Readings in English Transformational Grammar*. (Eds. Roderick Jacobs and Peter S. Rosenbaum) Boston, Mass.

___________ and M. Halle. 1968. *The Sound Pattern of English*. New York, Harper and Row.

Corstius, H. Brandt. 1970. *Exercises in Computational Linguistics*. Amsterdam.

Dougherty, Ching Yi. 1964. "The Lexeme *DE* as a Syntactic Marker". Project for Machine Translation, UC Berkeley. Berkeley, California.

Fillmore, Charles. 1968. "The case for case", in *Universals in Linguistic Theory*. (Eds. Emmon Bach and Robert Harms) New York, Holt, Rinehart and Winston, 1-88.

___________. 1970. "Types of Lexical Information", in *Studies in Syntax and Semantics*. (Ed. F. Kiefer) Holland, D. Reidel Publishing Company. 109-137.

Ginsburg, S. and H. G. Rice. 1962. "Two Families of Languages Related to ALGOL", *JACM*, 9:3,350-371.

Gruber, Jeffrey S. 1967. "Functions of the Lexicon in Formal Descriptive Grammars", Technical Memorandum TM-3770/000/00, System Development Corporation, Santa Monica, California.

Jakobson, Roman. 1957. "Boas' view of Grammatical Meaning", in *American Anthropologist*, 61:5, 144.

Jesperson, Otto. 1924. The Philosophy of Grammar. New York.

Lakoff, George. 1965. On the Nature of Syntactic Irregularity. Cambridge, Mass., The Computational Laboratory, Harvard University.

___________. 1970. "Pronominalization, Negation, and the Analysis of Adverbs", in Readings in English Transformational Grammar. (Eds. Roderick A. Jacobs and Peter S. Rosenbaum) Boston, Mass. 145-165.

Lees, Robert B. 1960. Grammar of English Nominalizations. The Hague: Mouton & Company. Also supplement to IJAL, 12, of the Research Center in Anthropology, Folklore, and Linguistics, reissue, 1963.

Liu, Yong-Quan. 1958. "Introduction to Machine Translation", Zhong Guo Yu Wen 78 (December 1958) 575-578.

___________. 1959. "Issledovatel'skaya Rabota V Oblasti Mashinnogo Perevoda V Kitajskoj Narodnoj Respublike" (Research Work in the Field of Machine Translation in the Chinese People's Republic), Voprosy Jazykoznanija VI.5 (Moscow 1959) 102-104. Russian. [English Transl. in JPRS 1131-D].

___________, Zu-Shun Gao and Zuo Liu. 1962. "A Comparison of Old and New Programs of Russian-Chinese Machine Translation Systems", Zhong Guo Yu Wen (October 1962) 439-458.

Lu Zhi-wei. 1957. Han Yu de Gouci Fa (Chinese Morphology) Peking Ke Xue Cu Ban Se.

McCawley, J. D. 1968. "Lexical Insertion", Papers of the Fourth Regional Meeting, Chicago Linguistic Society.

___________. 1968. "The Role of Semantics in a Grammar", in Universals in Linguistic Theory. (Eds. Emmon Bach and Robert T. Harms) Holt, Rinehart and Winston, Inc., New York.

Nida, Eugene. 1964. Toward a Science of Translating. Leiden, E. J. Brill.

Postal, Paul. 1964. Constituent Structure: A Study of Contemporary Models of Syntactic Description. Indiana University Research Center in Anthropology, Folklore and Linguistics.

Quarterly Progress Report #3 June 1969, #15 December 1969, #7 Novermber 1970. Project on Linguistic Analysis Syntax Group. Berkeley, California. UC Berkeley.

Reifler, Irwin. 1962. Final Report to the NSF. Washington University.

Ross, John R. 1967. *Constraints on Variables in Syntax.* PhD. dissertation. M.I.T. (mimeographed).

Sapir, Edward. 1921. *Language: An Introduction to the Study of Speech*. New York, Harcourt, Brace.

Tesniere, Lucien.. 1959. *Elements de Syntaxe Structurale*. Paris. [14].

Tosh, Wayne. 1965. *Syntactic Translation*. Mouton and Company. The Hague. Paris.

T'sou, Benjamin K. 1963. "Chinese Grammar I: Random Generation of Adjectival Modifier Constructions" (Internal Memorandum), Research Laboratory of Electronics, Machine Translation Group. M.I.T.

Vendler, Zeno. 1967. "Adjectives and Nominalizations", in *Linguistics in Philosophy*. Mouton.

____________. 1967. "Verbs and Times", in *Linguistics in Philosophy*. Cornell University Press. 97-121.

Wang, William S.-Y. 1964. "The Linguistic Basis of Translation". Paper given at the *Conference on Linguistics and Machine Translation*. Tokyo.

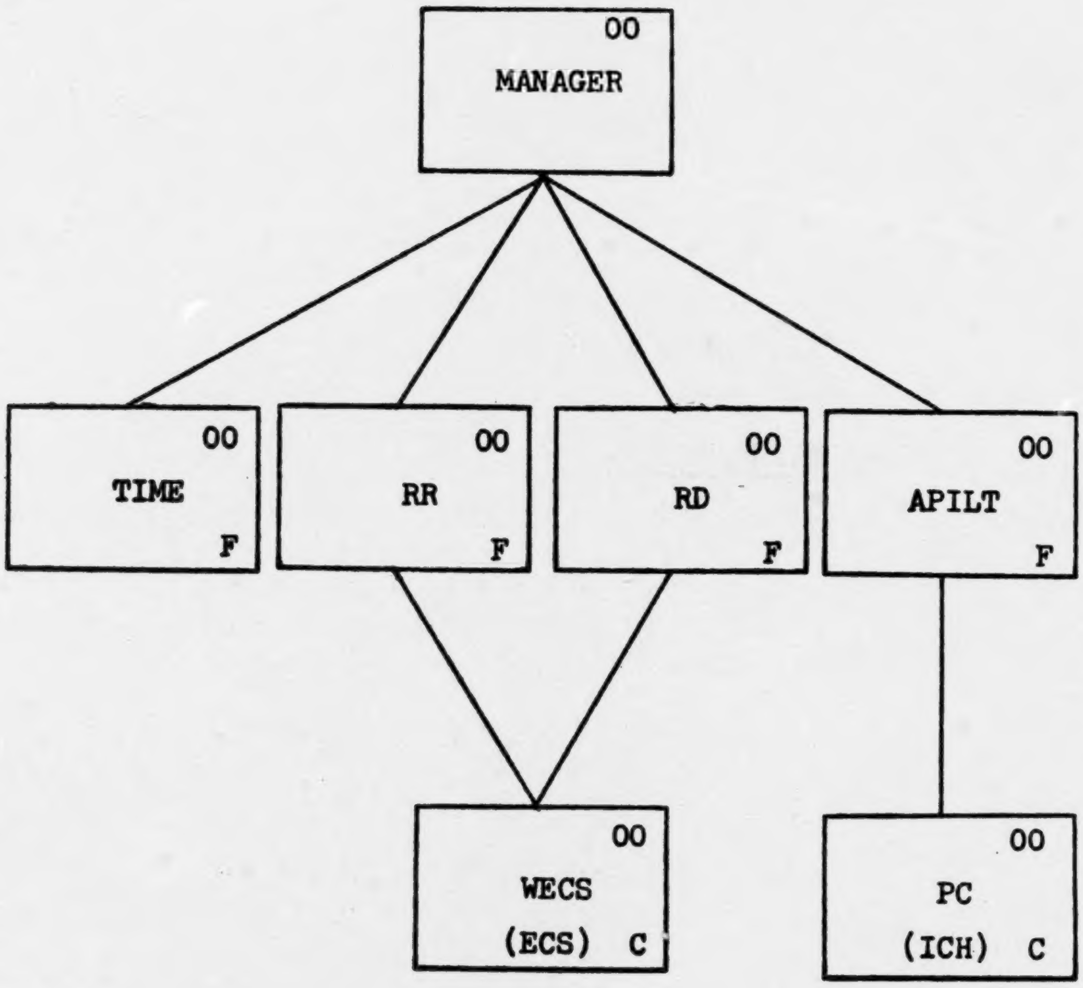
Yngve, Victor. 1960. "A Model and an Hypothesis for Language Structure", in *Proceedings of the American Philosophical Society*, Vol. 104, #5.

APPENDIX I

FLOWCHARTS

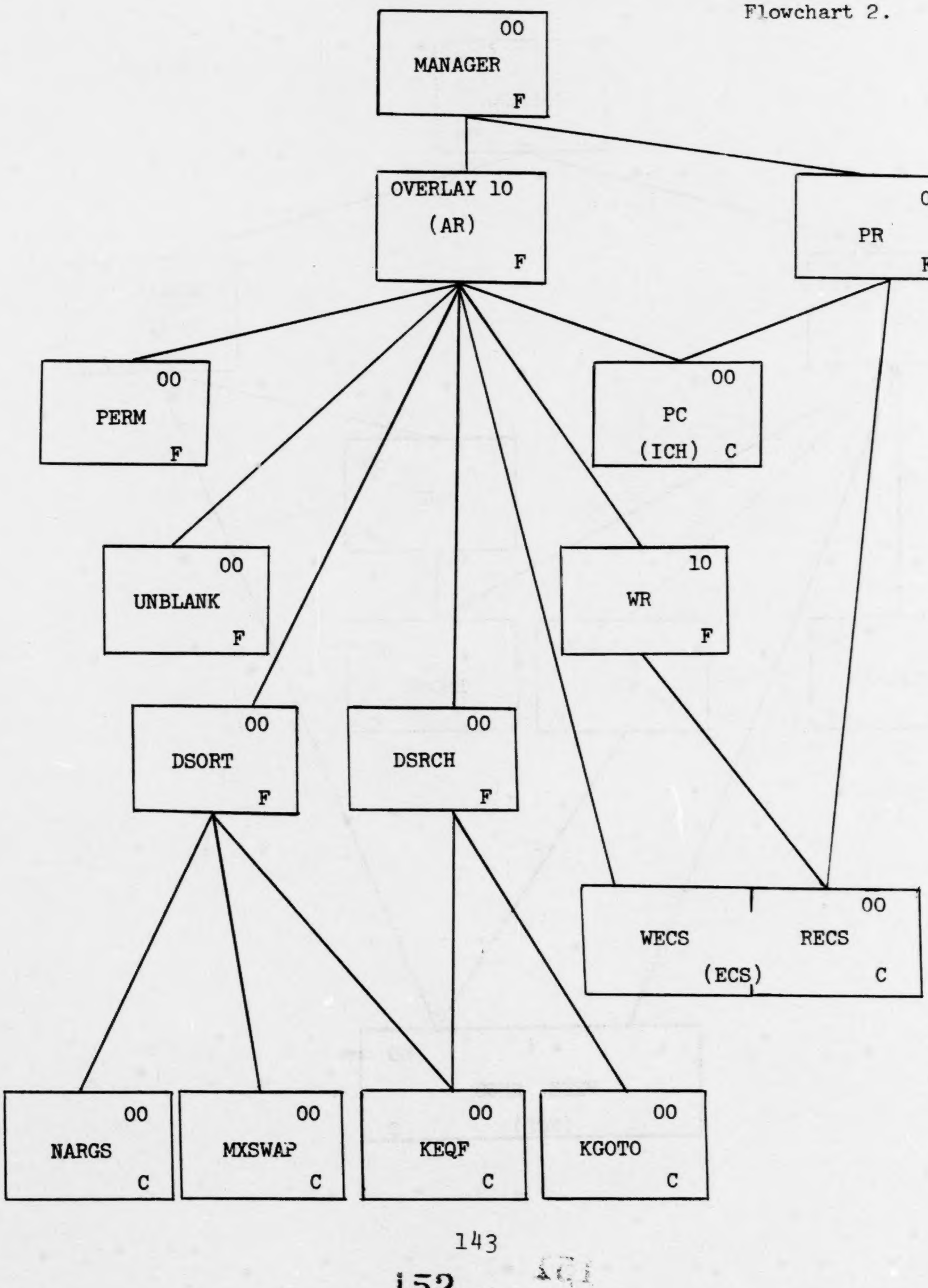
SUBROUTINE CALLS DURING THE PREPARATION PHASE OF A NORMAL RUN WITHOUT ADAPTATION OF RULES OR DICTIONARY

Flowchart 1.

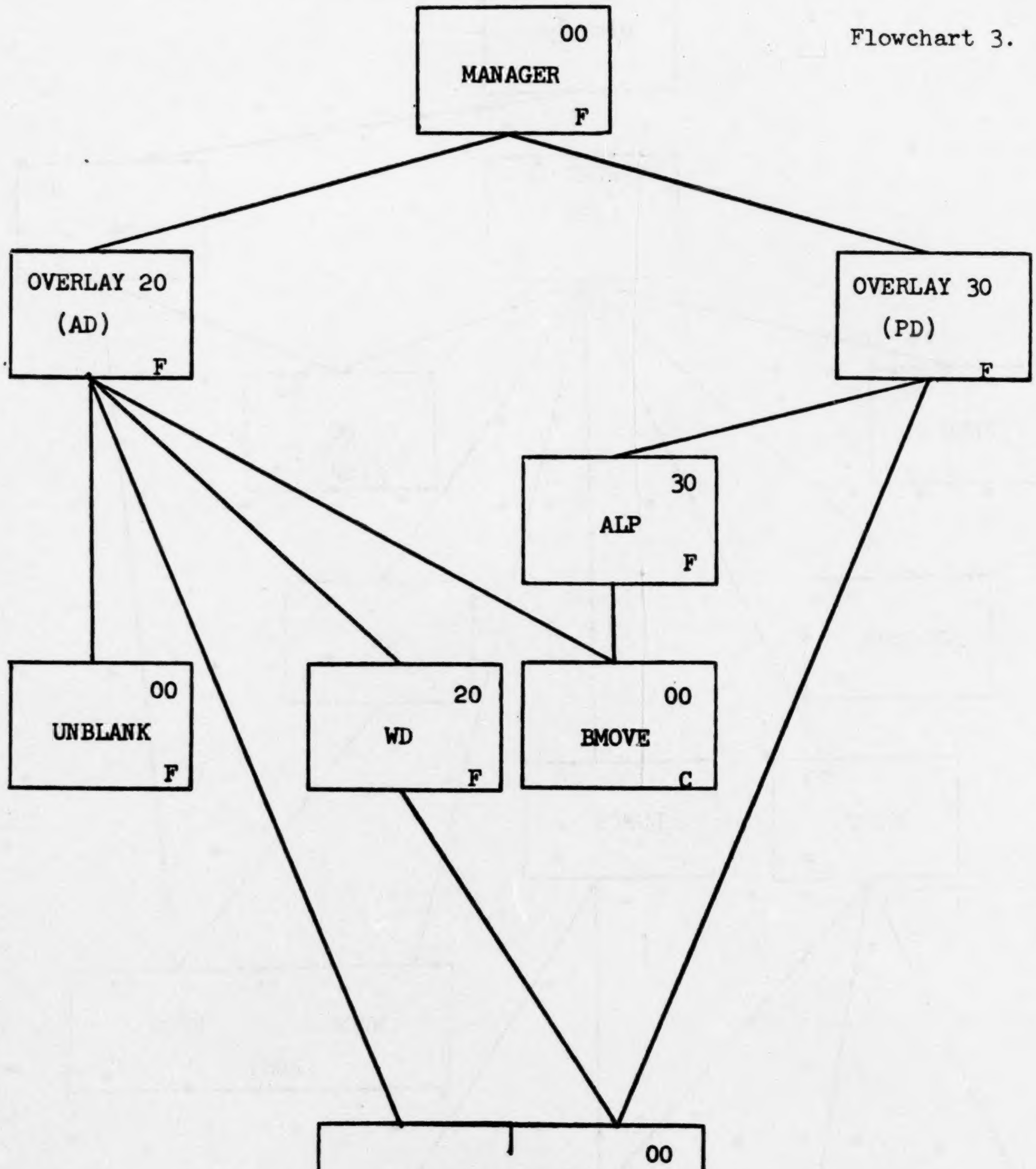

00
MANAGER
00
TIME
F
00
RR
F
00
RD
F
00
APILT
F
00
WECS
(ECS) C
00
PC
(ICH) C

SUBROUTINE CALLS DURING RULE ADAPTATION

Flowchart 2.


00
MANAGER
F
OVERLAY 10
(AR)
F
00
PR
F
00
PERM
F
00
PC
(ICH) C
00
UNBLANK
F
10
WR
F
00
DSORT
F
00
DSRCH
F
WECS
00
RECS
(ECS)
C
00
NARGS
C
00
MXSWAP
C
00
KEQF
C
00
KGOTO
C

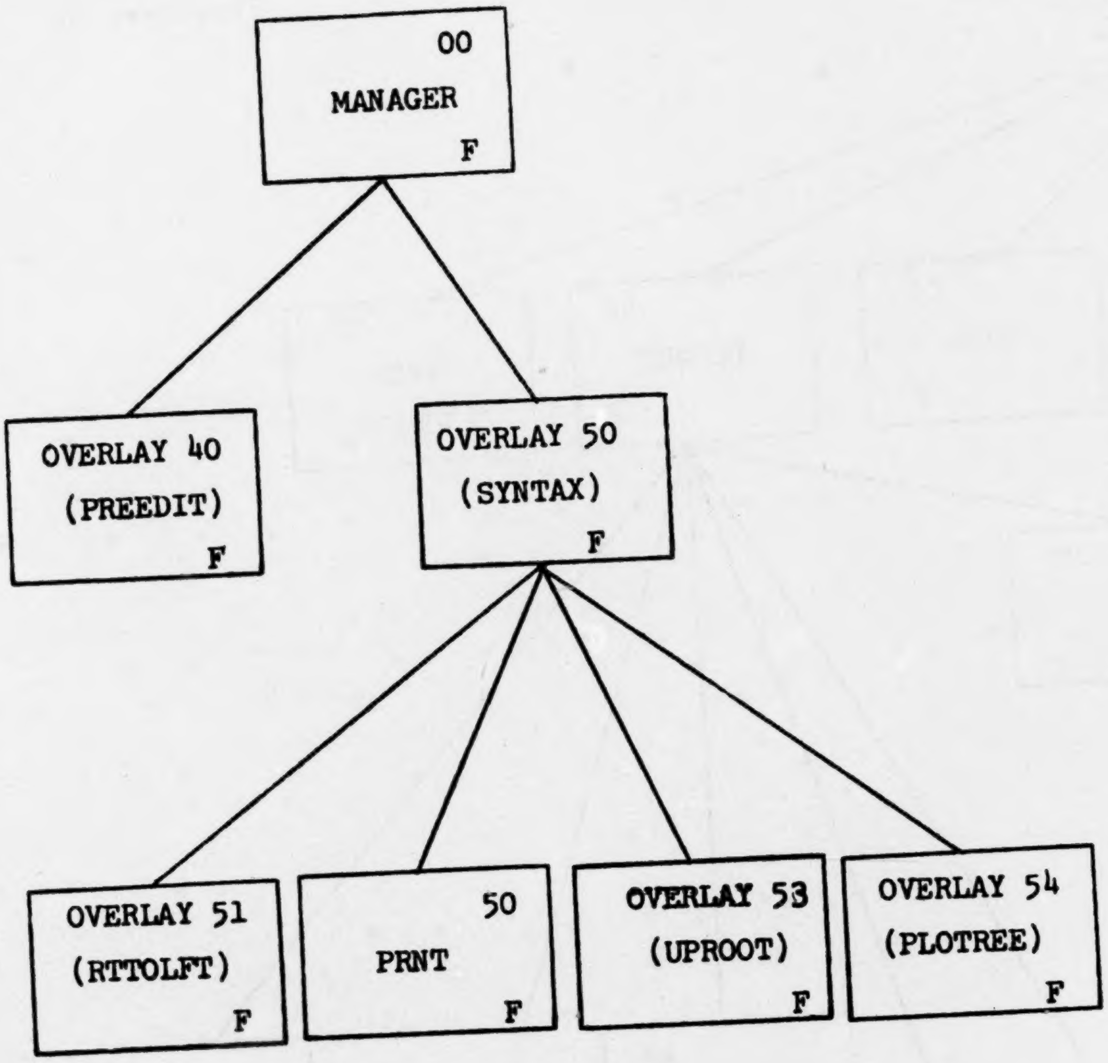
SUBROUTINE CALLS DURING DICTIONARY ADAPTATION


00
MANAGER
F
OVERLAY 20
(AD)
F
OVERLAY 30
(PD)
F
30
ALP
F
00
UNBLANK
F
20
WD
F
00
BMOVE
C
00
WECS RECS
(ECS)
C

Flowchart 3.

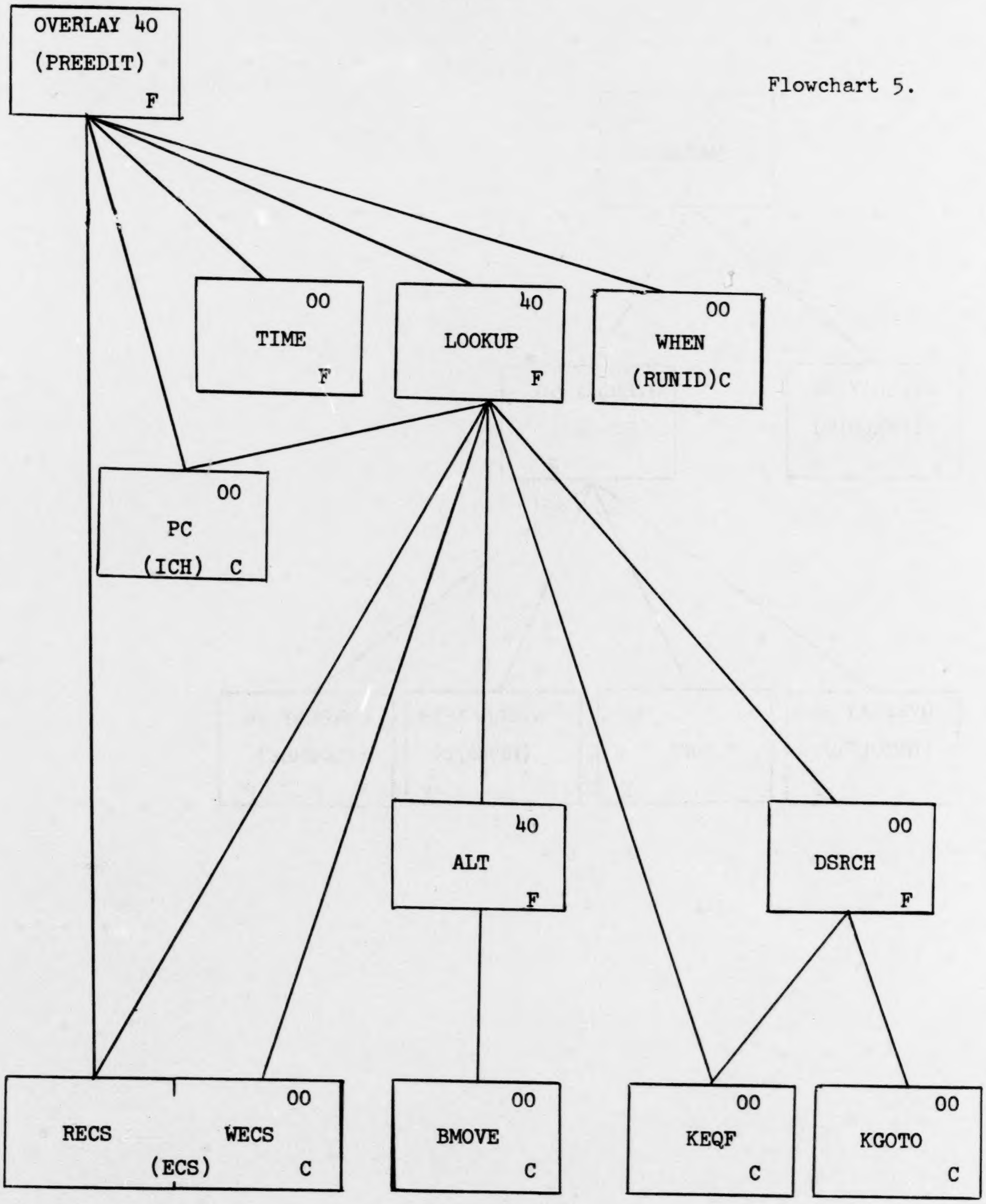
HIGH LEVEL CALLS DURING TEXT PROCESSING

Flowchart 4.


00
MANAGER
F
OVERLAY 40
(PREEDIT)
F
OVERLAY 50
(SYNTAX)
F
OVERLAY 51
(RTTOLFT)
F
50
PRNT
F
OVERLAY 53
(UPROOT)
F
OVERLAY 54
(PLOTREE)
F

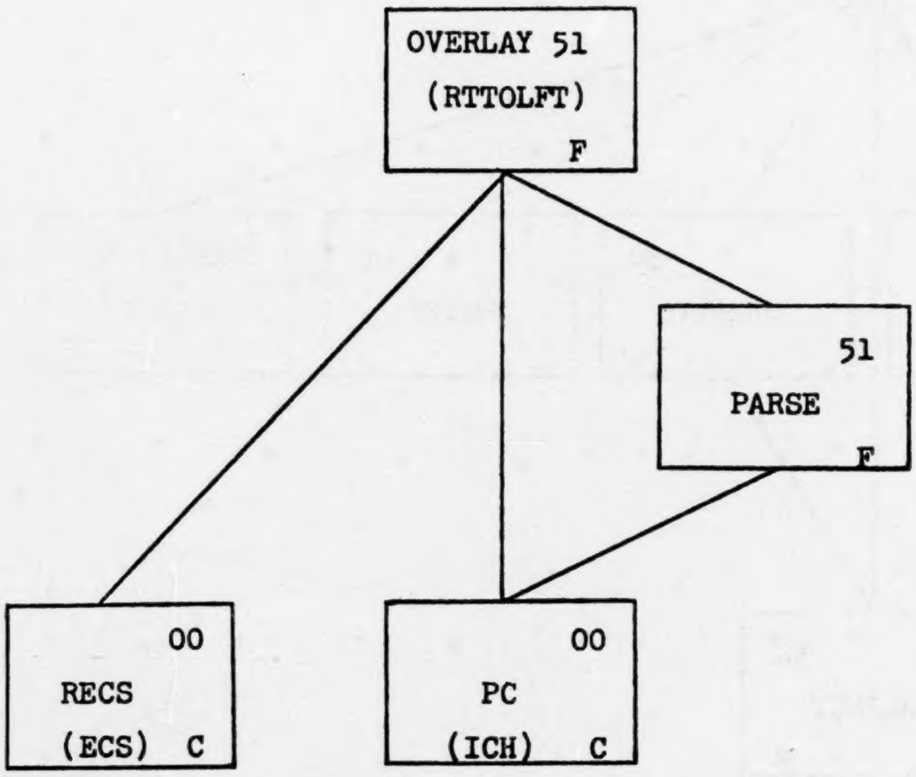
SUBROUTINE CALLS DURING TEXT PROCESSING UNDER OVERLAY 40

Flowchart 5.


OVERLAY 40
(PREEDIT)
F
00
TIME
F
40
LOOKUP
F
00
WHEN
(RUNID)C
00
PC
(ICH) C
40
ALT
F
00
DSRCH
F
RECS
WECS
00
(ECS)
C
00
BMOVE
C
00
KEQF
C
00
KGOTO
C

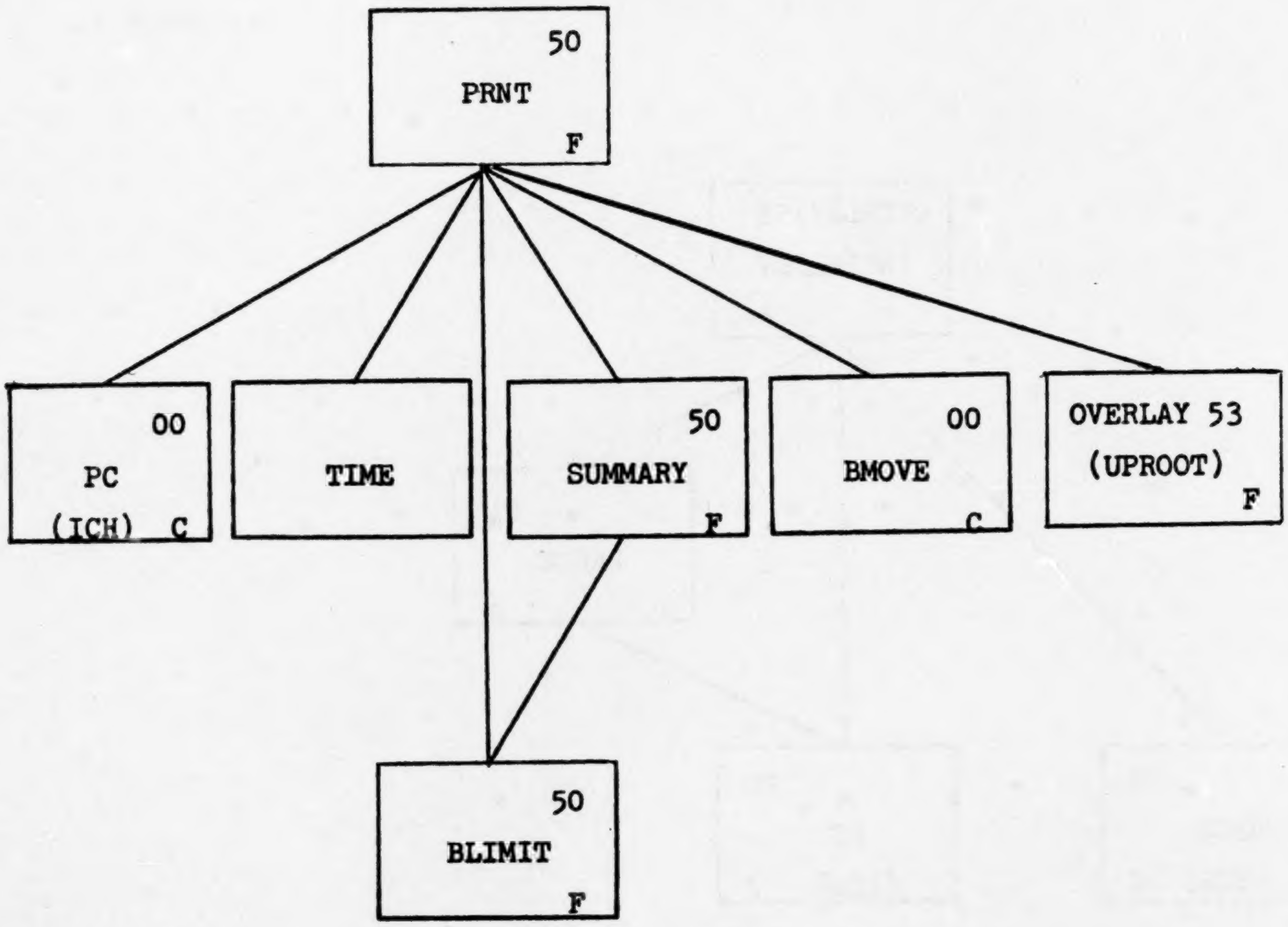
SUBROUTINE CALLS
DURING TEXT PROCESSING UNDER
OVERLAY 51

Flowchart 6.


OVERLAY 51
(RTTOLFT)
F
51
PARSE
F
00
RECS
(ECS) C
00
PC
(ICH) C

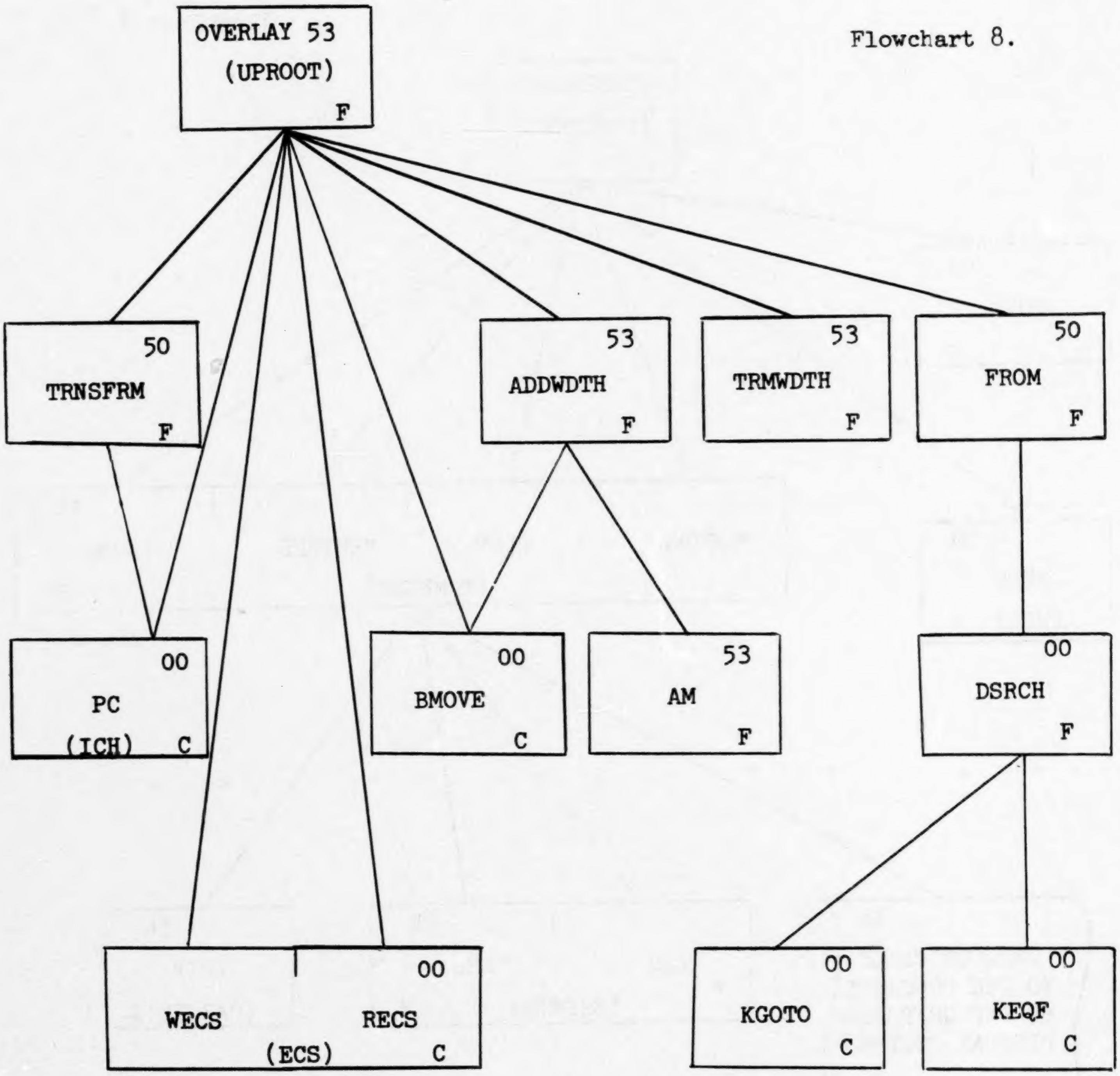
SUBROUTINE CALLS
DURING TEXT PROCESSING UNDER
SUBROUTINE PRNT

Flowchart 7.


50
PRNT
F
00
PC
(ICH) C
TIME
50
SUMMARY
F
00
BMOVE
C
OVERLAY 53
(UPROOT)
F
50
BLIMIT
F

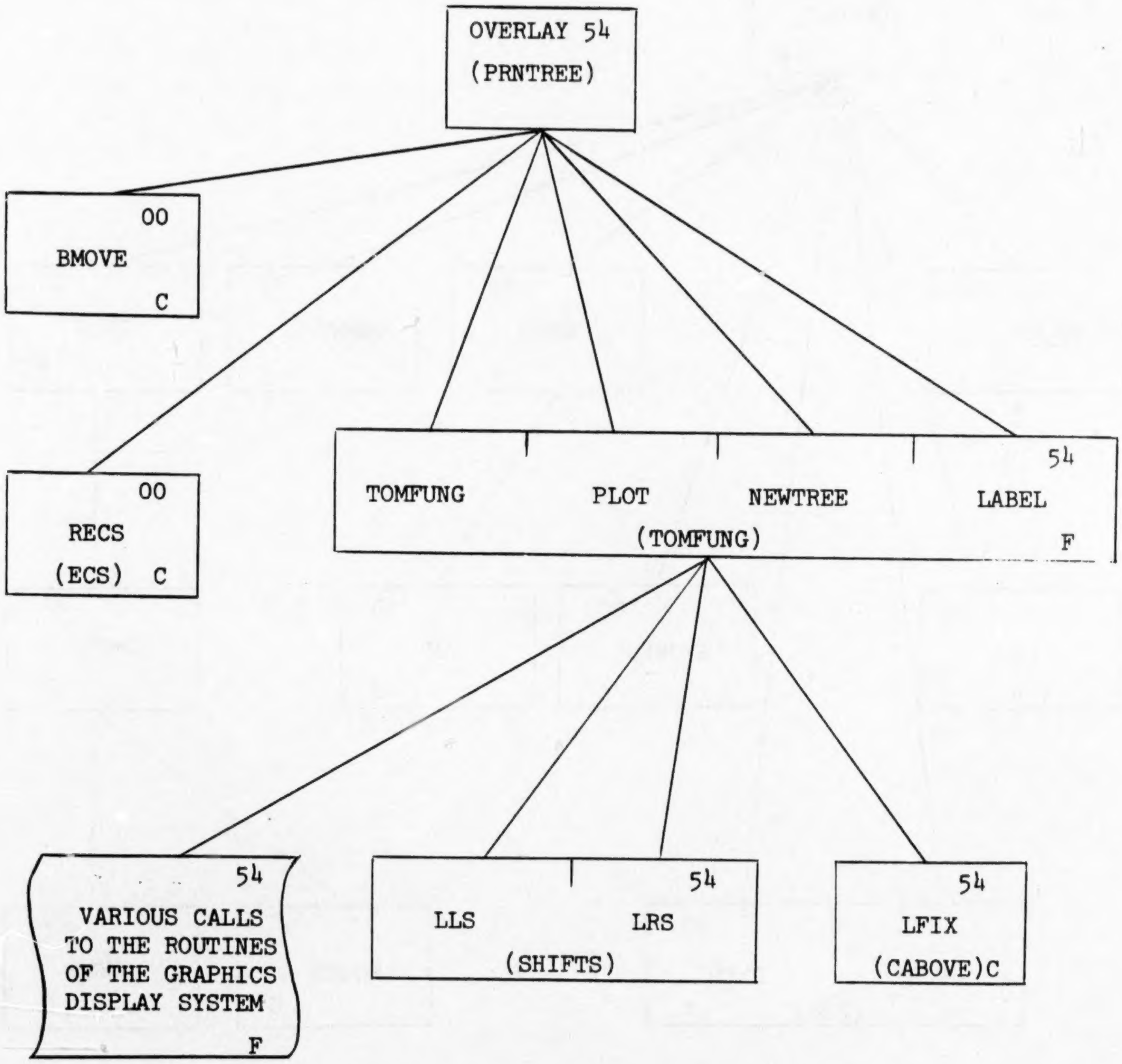
SUBROUTINE CALLS
DURING TEXT PROCESSING UNDER
OVERLAY 53

Flowchart 8.


OVERLAY 53
(UPROOT)
F
50
TRNSFRM
F
53
ADDWDTH
F
53
TRMWDTH
F
50
FROM
F
00
PC
(ICH)
C
00
BMOVE
C
53
AM
F
00
DSRCH
F
WECS
RECS
00
(ECS)
C
00
KGOTO
C
00
KEQF
C

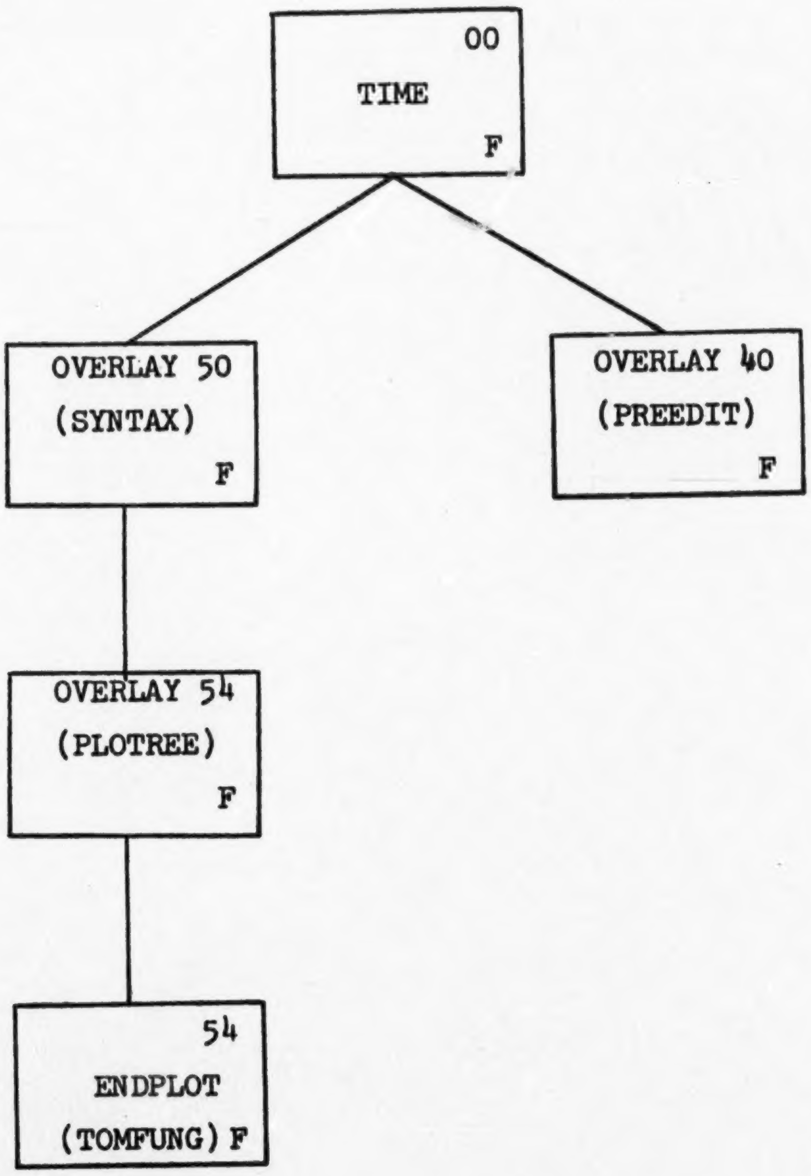
SUBROUTINE CALLS
DURING TEXT PROCESSING
UNDER OVERLAY 54

Flowchart 9.


OVERLAY 54
(PRNTREE)
00
BMOVE
C
00
RECS
(ECS) C
54
TOMFUNG
PLOT
NEWTREE
LABEL
(TOMFUNG)
F
54
VARIOUS CALLS
TO THE ROUTINES
OF THE GRAPHICS
DISPLAY SYSTEM
F
54
LLS
LRS
(SHIFTS)
54
LFIX
(CABOVE)C

SUBROUTINE CALLS AFTER INTERNAL TIME ESTIMATE HAS EXPIRED

Flowchart 10.


00
TIME
F
OVERLAY 50
(SYNTAX)
F
OVERLAY 40
(PREEDIT)
F
OVERLAY 54
(PLOTREE)
F
54
ENDPLOT
(TOMFUNG) F

APPENDIX II

SAS DOCUMENTATION

Name and Type: Program MANAGER in overlay (0,0)

Function: MANAGER is the resident code to manage the overlays of the SYNTAX ANALYSIS SYSTEM.

Called By: (MANAGER is the main program of the system and is not called.)

Calls: TIME, RR, PR, RD, and APILT, as well as overlays (1,0), (2,0), (3,0), (4,0) and (5,0)

Reads: MANAGER reads the LEAD card (on SCOPE file INPUT). See LEAD CARD PARAMETER TABLE.

Writes: nothing

Parameters: none

General Description: MANAGER reads the lead card, calls TIME, sets up some masks, makes the appropriate calls to read or create the adapted rules, dictionary, and interlingual transformations, and then once for each sentence calls overlay (4,0) and overlay (5,0), until either the text or the internal time estimate is exhausted.

Comments: MANAGER contains all I/O-buffers; the global common blocks FRSTREE, DIM, TRANSF, CAMBIG, STATUS, TERM, PARADIS, TFUNGBS, ID, INN, POINTER, TEXT, MASKS, TYME and SENSTAT are within its fieldlength also.

LEAD CARD PARAMETER TABLE

The following Parameters are read from the Lead Card:

Card Column	Parameter	Value	Action
1	Rule Processing (see MANAGER)	0	Read adapted rules
		1	Read and print adapted rules
		2	Read, adapt, and print rules
		3	None
2	Dictionary Processing (see MANAGER)	0	Read adapted dictionary
		1	Read and print adapted dictionary
		2	Read, adapt and print dictionary
		3	None
3	Constitute table (see PRNT)	0	None
		1	Print constitute table
4	Partial trees (see PRNT)	0	None
		1	Print partial trees of unparsed sentences
		2	Print partial trees of all sentences
5	Commas and Semicolons (see PREEDIT)	0	include C's and S's
		1	break on C's and S's
		2	skip C's and S's
6	English (see SYNTAX)	0	Do Chinese trees only
		1	Also do English trees
8	Chinese output (see UPROOT)	0	Neither print nor plot
		1	Print only
		4	Plot only
		5	Print and plot
9	English output (see UPROOT)	0	Neither print nor plot
		1	Print only
		4	Plot only
		5	Print and plot
11-14	Internal time limit in octal seconds		

Name and Type: Subroutine APILT in overlay (0,0)

Function: APILT adapts and prints the interlingual transformations.

Called By: MANAGER

Calls: PC (entry point of subroutine ICH)

Reads: interlingual transformations (card images on SCOPE file INPUT) one per card until a card with a 9 in column 1 is encountered

Writes: Echo prints each transformation as it is read on SCOPE file OUTPUT.

Parameters: none

General Description: The ten parameters specifying the transformation are read from columns one to eleven as integers, the 8th parameter being two digits, the others one digit. The twelfth column contains a character whose octal representation serves as the index into array TT in common block TRANSF of the word into which the ten parameters are packed for use by subroutine TRNSFRM.

Comments: APILT will soon be modified to handle insertions.

Interlingual Transformation Parameter Table

Card Column	
1	Input daughter position of output daughter 1
2	Input daughter position of output daughter 2
3	Input daughter position of output daughter 3
4	Input daughter position of output daughter 4
5	Input daughter position of output daughter 5
6	Input daughter position of output daughter 6
7	Input daughter position of output daughter 7
8 & 9	(Reserved for pseudo dictionary address of insertion)
10	Net number of daughters deleted
11	Single character address symbol for transformation

Name and Type: Subroutine BMOVE in overlay (0,0)

Function: moves a sequence of bits from a word or vector to another word or vector

Called By: AD, ALP, ALT, PRNT, UPROOT, ADDWDTH, PLOTREE

Calls: nothing

Reads: nothing

Writes: nothing

Parameters: 1 - the number of bits to be moved

2 - the source word (vector)

3 - the bit position in P2 at which the transfer is to start

4 - the destination word (vector)

5 - the first bit position in P4 into which transfered bits are to be placed

General Description: BMOVE gives FORTRAN access to bit manipulation and transfer best done in assembly language. Looping occurs only when bits from one vector are transfered to another.

Comments: Bit positions in vectors are not addressed by giving word number and bit number. Instead the convention that bits in word 1 are numbered 1-60, bits in word 2 are numbered 61-120, etc. is used.

Name and Type: Subroutine DSORT in overlay (0,0)

Function: sorts a matrix on the first row by the collating order of the 6400 character set; up to ten additional vectors can be carried along in the sort.

Called By: AR

Calls: NARGS, KEQF, MXSWAP

Reads: nothing

Writes: nothing

Parameters: 1 - the matrix to be sorted

2 - the number of columns in this matrix, i.e., the number of entries

3 - the number of rows in this matrix

4 - 13 (optional) additional vectors to be carried in the sort

General Description: DSORT is an in-core exchange sort using the Shell sort algorithm. Briefly, the Shell algorithm proceeds as follows:

(a) find the highest power of 2 less than P2 and subtract 1, call it L.

(b) use the L^{th} column as a boundary about which columns whose indices differ by L are exchanged when pairwise out of order

(c) if 2L is less than the number of columns continue exchanging columns whose indices differ by L in a backwards chain when necessary

(d) set L = L/2 (fixed point); if L = 0 sort is complete; otherwise go to step (b) using the $(P2 - L)^{th}$ column as a boundary point.

Comments: DSORT does not use the FORTRAN matrix representation directly; the matrix structure is transmitted by P2 and P3. The number of vectors carried in the sort is determined by the number of arguments in the call of DSORT as determined by subroutine NARGS.

Name and Type: Subroutine DSRCH in overlay (0,0)

Function: DSRCH is a binary search routine designed for used with FORTRAN matrices.

Called By: AR, LOOKUP, FROM

Calls: KEQF, KGOTO

Reads: nothing

Writes: nothing

Parameters: 1 - matrix to be searched

2 - the number of entries in P1

3 - the number of rows in P1

4 - P1-index of a search goal successfully found

5 - the search goal

6 - a vector in which the entire row for a goal successfully found is stored for return

7 - a branch address in case of an unsuccessful search

General Description: If P2 is less than zero, DSRCH assumes P1 is sorted from high to low on first words of each column; if P2 is greater than zero a low to high order is assumed.

KEQF is used for comparing entries in the search.

Briefly, the search proceeds as follows:

(a) find middle of P1 and compare with P5 (the goal of the search); if not equal determine which half of P1 NAME could be in;

(b) then find middle of that half and compare, if not equal, find half in which goal could be.

(c) iterate on (b) until either NAME is found or failure is assured.

Upon failure one branches to P7 in the calling routine. If a search for P5 is successful P6 contains the index of the entry in P1.

Comments: Immediate failure return occurs when P2 (the number of entries) ≤ 1 or P3 is ≤ 0.

DSRCH does not use the FORTRAN matrix structure; calculations of the memory position of matrix entries are done explicitly.

Name and Type: Function KEQF in overlay (0,0)

Function: Compares fixed-point members to determine the greater using the convention that any non-zero quantity is greater than zero.

Called By: DSORT, DSRCH

Calls: nothing

Reads: nothing

Writes: nothing

Parameters: 1 - a 60-bit fixed point number

2 - another 60-bit fixed point number

General Description: KEQF returns values as follows using 60-bit integer comparisons; if

$P2 > P1$ or $P2 = 0$ -1

$P2 = P1$ gives 0

$P2 < P1$ or $P1 = 0$ 1

Comments: The convention that zero is less than any non-zero quantity is used because KEQF is used to make lexicographic comparisons on the internal representation of alphanumeric material for DSORT, a sort program and DSRCH, a binary search program.

Name and Type: Subroutine MXSWAP in overlay (0,0)

Function: Exchange a column of one matrix for a column of another (they can be the same).

Called By: DSORT

Calls: nothing

Reads: nothing

Writes: nothing

Parameters: 1 - position in first matrix (considered as a vector) for the start of an exchange

2 - position in second matrix for the start of the exchange

3 - the number of rows in the matrix (= the number of words in each column)

General Description: MXSWAP exchanges the P3 words starting at P1 with the P3 words starting at P2.

Comments: MXSWAP is coded in COMPASS. It does not make use of the FORTRAN matrix representation. Before making a call to MXSWAP parameters 1 and 2 have to be calculated using the column indices of each matrix and the number of rows.

Name and Type: Function NARGS in overlay (0,0)

Function: When called from a FORTRAN routine R determines the number of arguments specified in the current call of R.

Called By: DSORT

Calls: nothing

Reads: nothing

Writes: nothing

Parameter: 1 - NARGS is called with a dummy parameter to insure its recognition as a function during compilation of a calling program.

General Description: This routine uses CDC RUN FORTRAN subroutine linkage conventions. It traces back through linkage words to find a linkage word in the routine (say P) whose current call (say R) called NARGS. The linkage word in P specifies the number of arguments in the current call of R.

Comments: NARGS is coded in COMPASS.

Name and Type: Subroutine PC (an entry point to ICH) in overlay (0,0)

Function: extracts a character or sequence of characters from a memory word and places them in another

Called By: AR, AD, APILT, PREEDIT, LOOKUP, TRNSFRM, PRNT, SUMMARY, RTTOLFT, PARSE, UPROOT, PLTREE

Calls: nothing

Reads: nothing

Writes: nothing

Parameters: 1 - the 60-bit word from which characters are to be extracted and moved

2 - the starting character position in P1 of the (sequence of) characters to be moved

3 - the 60-bit word in which characters moved are to be deposited

4 - the starting character position for placement of characters in P3

5 - the number of characters to be transfered

General Description: PC gives access from FORTRAN programs to the shifting and masking necessary for accessing and transfering packed characters.

Comments: PC is coded in COMPASS.

Name and Type: Subroutine PERM in overlay (0,0)

Function: Permutes columns of an array.

Called by: AR

Calls: nothing

Reads: nothing

Writes: nothing

Parameters: 1 - array name

2 and 3 - dimensions of array

4, 5, 6, 7, 8, 9, 10 input column numbers of output columns of the array.

General Description: PERM permutes up to seven columns of an array of words. Here a column is the set of all array elements having the same first coordinate.

Comments: It is used in the rule adaptation phase (AR) in order to permute the grammar codes in each rule.

Name and Type: Subroutine PR (Prints Rules) in overlay (0,0)

Function: This routine accesses a table grammar codes, incidental rule information, and a table of grammar rules and prints the information in each.

Called By: MANAGER

Calls: RECS (an entry point of ECS) and PC (an entry point of ICH)

Reads: nothing

Writes: the information described above on the SCOPE OUTPUT file

Parameters: none

General Description: The table of grammar codes and the incidental rule information are printed directly from the COMMON blocks in which they appear. The rule table appears in adapted form in ECS when PR is called. It is transferred to core using RECS; using PC, references to the grammar code table by index are unpacked from the one word representation of the rules; print representations of the grammar codes in each rule are then accessed via index and printed.

Comments: The incidental rule information is: (1) NG, the number of grammar codes, (2) NR, the number of rules, (3) IRV, the rule version, (4) IRD, the date of rule adaptation, and (5) IRR, the tape reel on which adapted rules are stored. In adapted form a grammar code in a rule is denoted by an index which points to the entry for that grammar code in the grammar code table. That entry includes the BCD representation of the grammar code.

Name and Type: Subroutine RD (Read Dictionary) in overlay (0,0)

Function: RD reads dictionary information from FORTRAN Logical tape I into core and transfers the information to ECS (Extended Core Storage).

Called By: MANAGER

Calls: WECS (an entry point of ECS)

Reads: FORTRAN logical tape number I

Writes: nothing

Parameters: (1) I, the number of the FORTRAN logical tape from which dictionary information is to be read.

Comments: Dictionary information is read and transferred in blocks of 2000 words.

Name and Type: Subroutine RECS (an entry point of ECS) in overlay (0,0)

Function: transfer information from ECS (Extended Core Storage) to core

Called By: PR, AR, WR, WD, PD, PREEDIT, LOOKUP, RTTOLFT, UPROOT

Calls: nothing

Reads: nothing

Writes: nothing

Parameters: 1 - the absolute address of the first word in core to which transfer is to be made

2 - the absolute address in ECS of the first word to be transferred

3 - the number of words to be transferred

4 - a failure pointer; returns (a) 0 if there is no parity error in the transfer, (b) (if positive) index of word at which parity error occurs when that index is found exactly, and (c) (if negative) the absolute value of P4 is the index of a word within three words of where the parity error occurs.

General Description: RECS evaluates P1, P2, and P3 and attempts a transfer. If the transfer is successful P4 is set to 0. If not, five rereads are attempted. If the parity error persists a binary search strategy of rereads is performed bracketing down to the offending word. P4 is set according to the success of the search.

Comments: RECS is written in COMPASS to give FORTRAN access to the COMPASS instruction RE which performs the transfer.

RECS and WECS are both entry points of ECS but are logically independent.

Indices mentioned in the description of P4 are taken with respect to the length of the transfer.

Name and Type: Subroutine RR (Read Rules) in overlay (0,0)

Function: This routine reads grammar codes, incidental rule parameters, and grammar rules. The rules are transferred to ECS (Extended Core Storage).

Called By: MANAGER

Calls: WECS (an entry point of ECS)

Reads: logical tape number 1 (rule and grammar code information)

Writes: nothing

Parameters: none

Comments: This routine consists entirely of fortran READ'S and a call to WECS which transfers material read into ECS.

Name and Type: Subroutine RUNID (Entry when) in overlay (0,0)

Function: format and retrieve results of calls on the system to give the time-of-day clock and calendar

Called By: PREEDIT

Calls: nothing

Reads: nothing

Writes: nothing

Parameters: 1 - word to return the date
2 - word to return the time

General Description: WHEN sets up the gibberish bit-strings necessary to ask for the time and date, asks a PP to fetch each in turn (hanging in recall until the PP complies), then formats the returned information--inserting colons and blanks, and turning the numerical date into one containing an alphabetic month. (WHEN does its own testing of the PP status bits, using its own macro RECAL in place of the compass library macro RECALL.)

The two words which return the result contain display code, of the form indicated by the following examples:

p1 - ƀ19:03:57ƀ

p2 - ƀ08ƀOCTƀ70

The time is based on a twenty-four hour clock; the month is always reduced to a three-character abbreviation. For most purposes, this form returned is suitable for printing out directly in two A10 formats.

Comments: This routine is coded in COMPASS, and is wholly machine-dependent; it involves crucially the CDC conventions for system requests and formats, and will have to be completely different in any other environment.

It should be mentioned that, for runs of the SAS, the combination of the date and time (nearest second) is certainly a unique identifier. For this reason, the data accessed by WHEN are intended to become part of the master key for sentences in test runs, as well as being part of the run identification.

Name and Type: Subroutine TIME in overlay (0,0)

Function: Monitors SAS "internal" time limit, and prints out elapsed time (in seconds and milliseconds) between any two successive calls.

Called by: MANAGER, PREEDIT, PRNT

Calls: SECOND, SYNTAX, PREEDIT

Reads: nothing

Writes: SCOPE fileset OUTPUT

Parameters: none

General Description: The current time is obtained by a call to the system clock, which is accessed by the system routine SECOND. If the "internal" time limit is not exceeded, processing continues. When time limit is exceeded, a message is printed and a status word is set, according to which plotting files will be closed and ambiguity tables will be printed as well as a count on the number of parsed and not parsed sentences.

Comments: It is nearly impossible to predict accurately how long processing of an unfamiliar text will take, yet if execution of an SAS run is abnormally terminated by a "time limit" interrupt it is not possible for SAS to regain control--which it must have so as to skip through the overlay structure closing files, and printing summary information which has been accumulating (the loss of which would reduce the value of the run considerably).

A solution to this problem is embodied in TIME: TIME is provided with an "internal" time limit--200_8 seconds less than that known to the system. Each time control is returned to the resident "MANAGER" overlay, the elapsed time is compared to the internal limit, and if the limit is exceeded a series of calls through the higher overlay structure is initiated. Each overlay begins with a block of code which interrogates the global status-word to decide whether it has been called for regular processing, or just for end-of-run action. [see flowcharts]

Name and Type: Subroutine UNBLANK in overlay (0,0)

Function: Converts blanks to octal zero's

Called By: AR, AD

Calls: nothing

Reads: nothing

Writes: nothing

Parameters: 1 - integer array
            2 - size of array

General Description: UNBLANK examines each 6 bit character position in the given array. Whenever it encounters a character position whose value is octal 55 (internal BCD blank) it resets the value to octal 00.

Comments: The removal of blanks is usually to create uniformity and proper collating sequence to facilitate comparison and sorting.

Name and Type: Subroutine WECS (an entry point of ECS) in overlay (0,0)

Function: Transfer information from core into ECS (Extended Core Storage)

Called By: RR, RD, FRECURS, AR, AD, LOOKUP, UPROOT

Calls: nothing

Reads: nothing

Writes: nothing

Parameters: 1 - the absolute address of the first word in core to be transferred

2 - the absolute address in ECS of the first word to which transfers are made

3 - the number of words to be transferred

General Description: WECS evaluates P1 and P2 and performs the transfer; an error branch is taken when ECS is not ready and an error message is written.

Comments: WECS written in COMPASS, to give FORTRAN access to the COMPASS intrusion WE which performs the transfer.

WECS and RECS are both entry points of ECS but are logically independent.

Name and Type: Program AR (Adapt Rules) in overlay (1,0)

Function: AR reads rules in card image form, binarizes all but unary rules, permutes the generated and generating grammar codes, prepares a printout representation and reference table, and prepares a core-image representation using indices from the reference table.

Called By: MANAGER

Calls: UNBLANK, PERM, DSORT, RECS, WECS, PC and WR

Reads: Rules in card image form from FORTRAN logical tape number 4

Writes: nothing

Parameters: none

General Description: Rules are read from FORTRAN logical tape number 4 in card image form. They are expected in the form: one generating grammar code and up to 6 generated codes. All non-unary and non-binary rules are binarized from left to right using unique new grammar codes for each new rule required. Rules are then permuted so that schematically a rule A goes to B C is stored as B C A. A reference table containing a hollerith representation of all grammar codes in the rules is constructed. A one-word packing of the rules is constructed using the indices of the grammar codes in the reference table as a representation of the grammar code. To each grammar code in the reference table a pointer to its first occurrence as a left constituent in a rule is added (the rules having been sorted so that rules having the same left constituent are grouped together).

Comments: The representation of the rules as packed indices is for compactness.

Name and Type: Subroutine WR (Write Rules) in overlay (1,0)

Function: WR stores a grammar code table, adapted grammar rules and associated information.

Called By: AR

Calls: RECS (an entry point of ECS)

Reads: nothing

Writes: FORTRAN logical tape number 1.

Parameters: none

General Description: The entire function of WR is to record rule information for future use. The grammar code table and associated information is written directly from core. Adapted rules are transfered from ECS and then written.

Comments: WR is called by AR to record the results of the rule adaptation which AR performs.

Name and Type: Program AD (Adapt Dictionary) in overlay (2,0)

Function: AD reads a dictionary whose entries are each 12 words long. These entries are segmented and stored in Extended Core Storage. Tables are prepared which provide quick access to each entry through that entry's first telecode.

Called By: MANAGER

Calls: UNBLANK, WECS, BMOVE, WD

Reads: FORTRAN logical tape number 2

Writes: nothing

Parameters: none

General Description: Dictionary entries contain (1) grammar code (1 60-bit word), (2) telecode sequence (3 60-bit words), (3) romanization (4 60-bit words), and (4) English gloss (4 60-bit words). It is convenient to store the grammar code and the telecode together and the romanization and English gloss together. These two parts of an entry are stored separately, but in parallel and similarly addressed tables.

Entries are grouped together by initial telecode. Addressing of the entries is by that first telecode. AD segments and stores the entries and prepares an address table IADDR, and a telecode substitution table.

Comment: The dictionary, which is on FORTRAN logical tape number 2, is presumed to be sorted by telecode.

Name and Type: Subroutine WD (Write Dictionary) in overlay (2,0)

Function: WD stores an adapted dictionary and incidental dictionary information.

Called By: AD

Calls: RECS (an entry point of ECS)

Reads: nothing

Writes: FORTRAN logical tape number 3

Parameters: none

General Description: The entire function of WD is to store the adapted dictionary for future use. The adapted dictionary is in ECS when WD is called. It is transferred in 2000 word blocks from ECS to core and written.

Name and Type: Program PD (Print Dictionary) in overlay (3,0)

Function: PD prints an adapted dictionary and associated tables, all of which are presumed to be in ECS (Extended Core Storage) at the time of call.

Called By:

Calls: RECS, ALP

Reads: nothing

Writes: dictionary information on the SCOPE OUTPUT file

Parameters: none

General Description: Dictionary information is stored in ECS in a special form (see documentation of AD). Using the address table provided two separate parts of each dictionary entry are accessed; they are then combined and printed.

Name and Type: Subroutine ALP in overlay (3,0)

Function: provide the ECS address and length of a character subdictionary

Called By: PD

Calls: BMOVE

Reads: nothing

Writes: nothing

Parameters: 1 - a numeric telecode (integer)

2 - returns adapted-subdic sequence number for the telecode in p1

3 - returns length of relevant entries

General Description: ALP is a routine for accessing IADDR, the core dictionary address table. Given the desired telecode in p1, ALP looks up in IADDR and returns the sequence number (p2) and the length--the number of entries beginning with this telecode (p3). The sequence number is thirteen bits, the length 7 bits; each is returned right-adjusted in its output parameter.

Comments: Table IADDR contains the addresses and lengths packed three to a sixty-bit word; the computation of ALP is to unpack the relevant twenty bits, and then to divide it between p2 and p3.

Name and Type: Program PREEDIT (in overlay 4,0)

Function: locate a string from the text, to be processed by the SAS, and perform some initial editing.

Called By: (MANAGER)

Calls: LOOKUP, TIME, PC, RECS

Reads: Text, card images, on SCOPE fileset INPUT

Writes: diagnostic messages on SCOPE ileset OUTPUT, plus a display of the text segments located

Parameters: none

General Description: PREEDIT scans the unedited input text looking for breaks indicating parsable units suitable for input to SAS--sentence-final punctuation, end-of-headings, etc. The options to break on commas and semicolons as well are selectable on the parameter card. PREEDIT removes extraneous punctuation (commas), all graphic information about type sizes and styles, and deletes everything within brackets and parentheses. Substitution of telecodes equivalenced in the dictionary is also performed.

The text string, thus cleaned up, is submitted to LOOKUP for lookup, and creation of a terminal table.

Comments: The current buffer for holding parsable units will hold strings up to two hundred characters (= telecodes) long.

Name and Type: Subroutine ALT in overlay (4,0)

Function: provide the ECS address and length of a character subdictionary

Called By: LOOKUP

Calls: BMOVE

Reads: nothing

Writes: nothing

Parameters: 1 - an actual telecode

2 - returns adapted-subdic sequence number for the telecode's entries

3 - returns length of relevant entries

General Description: ALT is a routine for accessing the table IADDR, the core table of dictionary addresses. ALT accepts a "raw"--alphameric--telecode in p1. It replaces a possible final letter with a zero, then converts the display code to integer format. That integer is used to access IADDR, to find the sequence number (p2) and number of entries beginning with that telecode (p3). The sequence number is thirteen bits, the length seven bits; each is returned right adjusted.

Comments: ALT presently does its BCD to Integer conversion with a DECODE statement.

Name and Type: Subroutine LOOKUP in overlay (4,0)

Function: find longest-matches from text in dictionary

Called By: PREEDIT

Calls: ALT, PC, RECS, WECS, DSRCH

Reads: nothing

Writes: error diagnostics on SCOPE file OUTPUT

Parameters:
1 - ECS base address of terminal table
2 - length of terminal table
3 - index of first telecode in text currently being looked up
4 - index of last telecode in text currently being looked up

General Description: LOOKUP begins by reading its basic address table IADDR in from ECS, and zeroing out the table to be filled with terminals. It then packs a comparison string of telecodes from the text, calls on ALT for indexes and brings in the relevant character sub-dictionary from ECS. This character dictionary is searched linearly, from the bottom up, and all longest matches are accepted (matching left to right). If no match results on a maximally long string (seven telecodes) progressively shorter matches are tried. On a success, a terminal (format below) is constructed and added to the table of terminals. This process is repeated through the text string. When LOOKUP is complete, the terminal table is written to ECS and return is made.

Comments: The format for terminals written by LOOKUP is a single sixty-bit word, divided into five twelve-bit fields. From high-order to low-order, these are:

1 - sentence position "from"
2 - binary zero's
3 - dictionary ID number of this entry

4 - address of (1) BCD and (2) rules entry point, for grammar code

5 - sentence position "to"

Name and Type: Program SYNTAX in overlay (5,0)

Function: SYNTAX manages overlays (5,0) through (5,4) which do the parsing, uprooting, transforming and plotting for each sentence. It also holds those large common tables which are used by more than one of these overlays, but not by overlay (4,0).

Called By: Control is passed to program SYNTAX by calls to overlay (5,0): once from program MANAGER in overlay (0,0) (which is executed once for each sentence), and once from subroutine TIME, also in overlay (0,0), which is executed only when the internal time estimate on the lead card has expired.

Calls: SYNTAX calls PRNT and calls overlays (5,1), (5,3) and (5,4) whose main programs are RTTOLFT, UPROOT, and PLOTREE, respectively.

Reads: nothing

Writes: nothing

Parameters: none

General Description: When SYNTAX is entered from TIME it calls overlay (5,4) to close the plotting. When SYNTAX is entered normally, i.e. from MANAGER, it calls overlay (5,1), subroutine PRNT, and overlays (5,3) and (5,4), according to their status word bits set in overlay (4,0), and to requests for plotting and transforming specified in the lead card. It then sets the status word for the next sentence and returns.

Name and Type: Subroutine BLIMIT in overlay (5,0)

Function: BLIMIT returns the index J of the last constitute for sentence position I, and the sentence position K of the first terminal to the left of I.

Called By: SUMMARY

Calls: nothing

Reads: nothing

Writes: nothing

Parameters: 1 - the input sentence position, I

2 - the constitute index J

3 - the sentence position K to the left of I

General Description: BLIMIT works by decrementing I successively until a sentence position K for which some terminal exists, is found. J is then set to one less than the index, B(K), of the first constitute for position K. The case I=1 is handled separately.

Name and Type: Subroutine FROM in overlay (5,0)

Function: Returns the sentence position J, of the Ith constitute.

Called By: SUMMARY

Calls: DSRCH

Reads: nothing

Writes: nothing

Parameters: 1 - input constitute index I

2 - output sentence position J

General Description: FROM uses DSRCH to research the B table. (B (N) is the first constitute having sentence position less than or equal to N.) When called by FROM, DSRCH returns the least J such that B(J) is greater than I, then J is incremented until B(J) changes.

Name and Type: Subroutine PRNT in overlay (5,0)

Function: PRNT prints the treetops, the B table, and the constitute table, and calls UPROOT and SUMMARY.

Called By: SYNTAX.

Calls: Overlay (5,3) (UPROOT), SUMMARY, TIME, BLIMIT, BMOVE, and PC (entry point of ICH)

Reads: nothing

Writes: output (described below) on SCOPE file OUTPUT

Parameters: none

General Description: PRNT first calls TIME for the elapsed time for parsing, then if the sentence was not parsed PRNT prints "not parsed" otherwise it reorders and prints the tree tops putting the preferred one first (1st choice SEN, 2nd choice NX5, 3rd choice IND2). If plotting is to be done the treetop count is then reduced to one so that only the best tree is plotted. Next the B table is printed, and if requested on the lead card, the constitute table is printed. If the sentence is not parsed then SUMMARY is called to print the breaks in parsing. If partial trees are requested on the lead card, overlay (5,3) is called and recalled to print them.

Comments: PRNT uses information from common blocks TEXT, POINTER, LYNE, WRKSPAC, GRCD, RULE, DICT, LDCD, DIM, CONST, SENSTAT, and TOP.

Name and Type: Subroutine SUMMARY in overlay (5,0)

Function: SUMMARY prints the breaks in an unparsed sentence.

Called By: PRINT

Calls: BLIMIT and PC (entry point of subroutine ICH)

Reads: nothing

Writes: prints the break table heading, and prints each span between breaks with the type of break and all grammar codes found for the span. An error message may also be printed if for some span the above grammar codes overflow their table, which is dimensioned 100. (all on SCOPE file OUTPUT)

Parameters: none

General Description: Beginning with sentence position one, SUMMARY makes a list of the grammar codes of all constitutes which begin with the current sentence position and span a maximal portion of text. Then the constitutes for the sentence positions of the rest of the span are examined to determine whether the break is absolute. The results are printed, and the sentence position immediately to the right of the maximal span becomes the current position. The cycle is repeated until the sentence is exhausted.

Name and Type: Subroutine TRNSFRM in overlay (5,0)

Function: TRNSFRM performs interlingual permutations and deletions.

Called By: UPROOT (during the uprooting and printing of the English tree)

Calls: PC (entry point of subroutine ICH)

Reads: nothing

Writes: two error messages (on SCOPE file OUTPUT): one for the attempted execution of undefined transformations, the other for incorrectly specified transformations with more than seven fields to be permuted.

Parameters: none

General Description: TRNSFRM applies the transformation indexed by the character following an asterisk in the penultimate character position of the printout representation of the grammar code of the current node. The transformation so indexed is applied at one level of indirection. That is, the pointers to the nodes are permuted or deleted, rather than the nodes themselves.

Name and Type: Program RTTOLFT in overlay (5,1)

Function: RTTOLFT "walks through" each sentence from right to left creating the constitutes for each terminal and calling PARSE for each sentence position.

Called By: SYNTAX (overlay (5,1)

Calls: PARSE, RECS, and PC (entry point of subroutine ICH)

Reads: nothing

Writes: prints two error messages on SCOPE file OUTPUT: one for ECS parity errors, the other for terminal table out of sequence

Parameters: none

General Description: RTTOLFT first brings in the rules and terminals from ECS. Then it zeros out the constitute table and initializes its pointers to walk through the sentence from right to left. At each sentence position it examines the terminal table and generates one terminal constitute for each terminal. It then calls PARSE for that sentence position and goes on to the next, repeating the cycle until position zero is reached and then returning control to SYNTAX in overlay (5,0).

Name and Type: Subroutine PARSE in overlay (5,1)

Function: PARSE creates all nonterminal constitutes, which begin at the current sentence position.

Called By: RTTOLFT

Calls: PC (entry point of subroutine ICH)

Reads: none

Writes: prints error messages on SCOPE file OUTPUT if the constitute table or the ambiguity table is full. The third error message, "PARSE 160", would indicate a bug involving the creation of the B table.

Parameters: none

General Description: PARSE begins by setting up some masks and initializing ILC, the index of the left candidate, to point at the fictional zeroth constitute for the current sentence position.

Then the outer loop is entered, incrementing the left candidate index and creating all possible constitutes for which the current left candidate has been used as left constituent. That this will happen is guaranteed by the absence of loops of Unary Rules, together with the finiteness of the list of possible right candidates for use with Binary Rules. Within the above loop the process for each left candidate is as follows:

First the list of rules (productions) in which the grammar code of the left candidate appears in left constituent position is determined.

Second, each of the Unary Rules on this list is used to create a new Unary constitute.

Third, the list of all constitutes spanning strings beginning immediately to the right of the string spanned by the left candidate is determined. This is the list of possible right candidates.

Fourth, an intermediate loop is entered incrementing IRC, the index of the right candidate, and creating all possible constitutes which have the current left and right candidates as left and right immediate constituents respectively. For each right candidate this is done by going through an inner loop incrementing the rule index, IR. For each value of IR, the grammar code in right constituent position is compared with the grammar code of the right candidate. If equal, a new binary constitute is constructed.

Whenever a new (unary or binary) constitute is formed a check is made to determine whether another constitute with the same grammar code spans the same string. If so, the old constitute is moved up to follow the new one, and in its old place, catching all references to it, a summarizing constitute is created, with the new and old constitutes as left and right immediate alternatives respectively. The new and old constitutes themselves are then rendered quiescent, and may not be used in future parsing. The grammar code common to these nodes is then looked up in the ambiguity table. If found, its counter is incremented and it is bubbled up into proper position by counter. If not found it is added at the end of the table with counter set to 1.

Any non-quiescent constitute whose grammar code does not go up via unary rules is recorded as a "treetop" (=root).

Comments: PARSE uses common tables LEADCD, TEXT, DIM, POINTER, CONST, CAMBIG, RULE, GRCD, and WRKSPAC. During the execution of PARSE, WRKSPAC contains the adapted grammar rules. A faster parser, using hash tables, and providing greater grammatical flexibility, is under design and will soon replace this one.

Name and Type: Program UPROOT in overlay (5,3) [Note: overlay 5,2 does not exist.]

Function: UPROOT "pulls" a tree out of constitute table by its root (also called treetop, since these trees are upside down).

Called By: SYNTAX, PRNT

Calls: TRNSERM, TRMWDTH, ADDWDTH, FROM and BMOVE, PC (entry points of subroutine ICH) and RECS and WECS (entry points of subrouting ECS)

Reads: nothing

Writes: prints the output trees and their headings, and two error messages, one for more than 499 'see aboves', and one for read parity error. A third error message would indicate that through a bug an empty constitute was to be used in creating the tree.

Parameters: none

General Description: UPROOT begins by seeting up its control words and masks, forming a clean copy of the constitute table in ECS, and zeroing out the space for new tree tables. Then UPROOT enters its main loop, through which it will pass once for each node in the tree it is uprooting from the constitute table. It traverses the tree in the order: top-left-right.

UPROOT begins with the tree top constitute whose index ITT is received through common block TOP. If the English tree is being uprooted then TRNSFRM is called after each node is processed and before determining which node to do next. After processing a non-terminal constitute UPROOT proceeds to its left immediate constituent or alternative via the index in the parent constitute. After processing a terminal constitute, UPROOT checks its tables of pseudo siblings, true siblings and parents, to find the lowest branch to the right. Pseudo constituents and pseudo alternatives resulting from the binary format of the constitute table are eliminated at this stage. Not to be confused with these pseudo nodes which are eliminated are the pseudo terminals which are in all ways like true terminals except that they are obtained from a second dictionary either through interlingual insertions or through special handling or names. Although UPROOT is designed to handle them, these pseudo terminals are not created by the present version of the system; they will be added soon.

Each time through the big loop, UPROOT first determines whether the current constitute is new or old. If it is a previously encountered constitute then it has already been expanded as a part of some other alternative and will this time be represented only by a note "see above" and the number of the expanded node. In other respects these "see above" nodes function as terminals - both in computing the X-coordinates and widths of nodes to be plotted, and in determining which constitute to process next, the "see above" nodes are handled the same as terminals.

Based on whether UPROOT was called with status word one, UPROOT does the Chinese or English tree. For whichever language it is doing, it checks the lead card parameter for that language, which specifies separately whether or not to print the tree, and whether or not to plot the tree. UPROOT should not be called when neither printing nor plotting is to be done for the language specified. If it were called in this case, then the tasks required both by printing and by plotting would be executed. These are the generation of the tables of tree nodes without their widths and plotting coordinates. If printing is requested it is this information that is printed. If plotting is requested, then TRMWDTH is called to compute the width of the terminals and "see aboves" and ADDWDTH is called to compute their X-coordinates, and to accumulate the widths and compute the X-coordinates for the non-terminal nodes.

The plotting itself is done later in overlay (5,4) from the tree node tables created by UPROOT.

Comments: The tree is created as two tables in common blocks WRKSPAC and ENTRY and will also be printed if requested on the lead card. If plotting is not requested some of the fields in the tables are not computed.

UPROOT uses common blocks: PLOTSIZ, STATUS, LEADCD, EYE, ENTRY, WRKSPAC, CONST, GRCD, TREESIZ and TOP.

Name and Type: Subroutine ADDWDTH in overlay (5,3)

Function: ADDWDTH computes the width and X-coordinate in character positions for each tree node to be plotted.

Called By: UPROOT

Calls: BMOVE, AM

Reads: nothing

Writes: prints an error message on SCOPE file OUTPUT if an unexpected zero node is encountered.

Parameters: none

General Description: ADDWDTH accumulates the widths of the lowest level preceding nodes (terminals and "see aboves") to get the X-coordinate of the current lowest level node, and then computes the width and X-coordinate of each higher node from those of its constituents.

Comments: ADDWDTH uses the treenodes which are in common blocks WRKSPAC and ENTRY.

Name and Type: Subroutine TRMWDTH in overlay (5,3)

Function: TRMWDTH returns 1 + the width in character positions of a string of non-blank characters.

Called By: TRMWDTH is called by UPROOT.

Calls: nothing

Reads: nothing

Writes: nothing

Parameters: TRMWDTH has three calling parameters:

1 - M, an array containing the string

2 - N, the number of words in M

3 - K, one plus the width

General Description: TRMWDTH begins at the right end of the given string, the tenth character position of word M(N), and scans leftward until it encounters a character which is neither blank nor octal zero. This is the Jth character in M(I). TRMWDTH then sets K = 10*I + J - 9, and returns.

Name and Type: Program PLOTREE in overlay (5,4)

Function: PLOTREE manages the plotting.

Called By: entered by calls to overlay (5,4) from SYNTAX in overlay (5,0).

Calls: BMOVE, RECS (entry point of ECS), PC (entry point of ICH) and entry points TOMFUNG, NEWTREE, PLOT, LABEL, and ENDPLOT of subroutine TOMFUNG. [A call to entry point SERIAL of TOMFUNG has been temporarily removed pending new record formats to make use of it.]

Reads: FORTRAN logical tape 41, the sentence labels

Writes: prints only an ECS read parity error message on SCOPE file OUTPUT

Parameters: none

General Description: If PLOTREE is called with the eighth bit on in the status word, signifying that the internal time estimate has expired, it calls ENDPLOT and returns. When called for the first time, PLOTREE begins by calling TOMFUNG. Otherwise it begins by calling NEWTREE. Then it calls PLOT once for each node, calls LABEL and returns.

Comments: uses the adapted dictionary in ECS as well as common blocks STATUS, TERM, PLOTSIZ, GRCD, WRKSPAC, ENTRY, FRSTREE and MASKS

Name and Type: Subroutine ENDPLOT (an entry point to TOMFUNG) in overlay (5,4)

Function: Carries out closing housekeeping on the file of plot commands, after all plotting has been completed.

Called By: PLOTREE

Calls: GDSEND (a part of the proprietary Graphic Display System)

Reads: nothing

Writes: FORTRAN logical tape 99

Parameters: ENDPLOT ignores its argument list.

General Description: ENDPLOT is simply a call to the GDS routine GDSEND, required for its own purposes. No plotting calls after a call to ENDPLOT can produce any output.

Comments: Through the device of multiple entry points, ENDPLOT shares the communications areas of TOMFUNG and the other plotting routines.

Name and Type: Subroutine LABEL (an entry point to TOMFUNG) in overlay (5,4)

Function: to label plotted parse trees with text sentence, grammar and dictionary version, and other identification

Called By: PLOTREE

Calls: TITLER (a part of the proprietary Graphic Display System)

Reads: nothing

Writes: FORTRAN logical tape 99 (an intermediate file)

Parameters: 1 - the label to be written. May take the form either of (a) a hollerith constant or (b) an array reference where the array containing the text is blank filled, last partial text word left-justified and blank-filled, first non-text word all octal zeroes.

General Description: LABEL will write up to 1060 characters per line vertically at the right-hand side of the paper, where the labels may be seen by unrolling the scroll of output a little way.

Successive calls to LABEL are automatically off-set further and further away from the plotting area; thus, LABEL may be called repeatedly for multiline titles.

Comments: Through the device of multiple entry points, LABEL shares the communications area of TOMFUNG and the other plotting routines.

Name and Type: Subroutine LFIX in overlay (5,4)

Function: Character manipulation for see-above's

Called By: PLOT

Calls: nothing

Reads: nothing

Writes: nothing

Parameters: 1 - see-above information (passed to PLOT as its parameter eight; see PLOT)

2 - grammar code reference for current node

3 - a three-word array, in which results are written by LFIX

4 - returns (a) -1 if a see reference is to be written; (b) 0 if a labeled grammar code is to be written for later reference; (c) 1 if no see-above treatment is required

General Description: LFIX begins by left shifting its p1 and testing against zero; if the word received contained 18 high-order zero bits, then return is made immediately taking no action.

If the zero test fails, then the shifted word is tested for being negative --that is, is the bit adjacent ot the former high-order 18 on? If it succeeds, the grammar code from p2 and the see-above reference number from p1 are joined--separated by a slash--thus labeling the grammar code. The results are returned in p3, followed by at least one zero word. The combination may exceed a single word, and LFIX will handle this.

If that is not what is wanted, then the next adjacent bit is tested; if off, return is made as in the first case mentioned above. If on, then a word is constructed from the letters "SEE", a blank, and the three-digit reference passed in p1; the result is written as the first word of p3, and return is made.

A record of which action is taken is returned in p4, in a convenient form for FORTRAN testing and branching.

Comments: The routine is coded in COMPASS, primarily for efficiency in the unpacking loop needed for labelling grammar codes.

Name and Type: Integer Function LLS in overlay (5,4)

Function: performs circular (logical) left shift of a 60-bit word

Called By: TOMFUNG

Calls: nothing

Reads: nothing

Writes: nothing

Parameters: 1 - word to be shifted
2 - number of bits left shift (decimal)

General Description: p1 is left-shifted p2 bits, circularly. If p2 is negative, shift is arithmetic right by ABS (p2) bits. Result is left in register X6, the RUN FORTRAN convention for function calls.

Comments: Routine is coded in COMPASS, and corresponds to a hardware facility in the order code of the CDC6400.

Name and Type: Integer Function LRS in overlay (5,4)

Function: performs arithmetic right shift of a 60-bit word

Called By: TOMFUNG

Calls: nothing

Reads: nothing

Writes: nothing

Parameters: 1 - word to be shifted

2 - number of bits right shift (decimal)

General Description: p1 is right-shifted p2 bits, end-off with copies of the sign bit shifted in for fill. If p2 is negative, shift is left circular by ABS (p2) bits. The result is left in register X6, the RUN FORTRAN convention for function calls.

Comments: Routine is coded in COMPASS, and corresponds to a hardware facility in the order code of the CDC6400.

Name and Type: Subroutine NEWTREE (an entry point to TOMFUNG) in overlay (5,4)

Function: initializes successive frames of a single plotting run

Called By: PLOTREE

Calls: NXTFRM (which is part of the proprietary Graphic Display System)

Reads: nothing

Writes: FORTRAN logical tape 99 (an intermediate file)

Parameters: 1 - number of character positions along the X-axis for the new tree frame (floating-point)

2 - number of levels for the new tree along the Y-axis (floating-point)

3 - returns 0.0 for successful initialization, or -1.0 if requested density of information is too great to be labeled successfully

General Description: NEWTREE closes the old plotting frame and moves to a new frame on the plotter paper. It resets paper margins which may have been changed by a labeling call. It then physically transfers control to the TOMFUNG entry point, and TOMFUNG initializes the next frame in the usual way.

If the second parameter is negative, then NEWTREE will close the old frame without initializing the following frame. After such a call, a call to TOMFUNG must be made explicitly before further plotting can be done.

Comments: Through the device of multiple entry points, NEWTREE shares the communications areas of TOMFUNG and the other plotting subroutines.

Name and Type: Subroutine PLOT (an entry point to TOMFUNG) in overlay (5,4)

Function: the basic plotting call; draws a line from one point to another, and labels the second point.

Called By: PLOTREE

Calls: LLS, LFIX (plus SLLILI, DLLILI, and TITLEG, which are part of the proprietary Graphic Display System)

Reads: nothing

Writes: FORTRAN logical tape 99 (intermediate file)

Parameters: 1 - X-coordinate of point 1, in character positions (floating-point)

2 - Y-coordinate of point 1, in levels (floating-point)

3 - X-coordinate of point 2, in character positions, (floating-point)

4 - Y-coordinate of point 2, in levels (floating-point)

5 - 0.0 for solid line; non-0.0 for dashed line

6 - Integer location in COMMON/GRCD/ of the display-code string which is the label for point 2. Each label is in one word, left-justified, blank-filled.

7 - Integer 0 for non-terminal nodes; if non-0, then COMMON/TERM/ contains 12 words of annotation for this terminal node--four words each of telecodes, romanization, and English

8 - A three-digit see-above reference number, plus information for handling it. If the high-order 18 bits of the word are octal zeroes, no see-above handling is wanted. If those 18 bits are non-zero, they are display code for the see-above reference. If the adjacent bit is on, then this node is to be labeled so that a see-above may refer to it; if the bit adjacent to that is on instead, then this node is to be a see-above node. The right

40 bits of the word are ignored.

9 - "from and to," the sentence positions dominated by the node to be labeled; two three-digit display code numbers. To is in the low-order 18 bits of the word, From in the adjacent 18 bits.

General Description: PLOT first checks to see if it has previously been initialized properly (by an acceptable call to TOMFUNG); if not, it returns immediately taking no action.

If all is well, PLOT draws a line from point one to point two, solid or dashed (dashed lines are used to indicate alternatives of ambiguities). The points are automatically offset by PLOT to avoid drawing over annotation, so the calling program may treat the points as real points.

The grammar code for the second point is centered and written, and the sentence positions dominated by this node are connected by a dash, centered, and written below. If see-above treatment is requested, either (a) a reference number is prefixed to the grammar code or (b) a see-above line is written (both handled by subroutine LFIX).

If this is a terminal node, another line is drawn to the line of terminals, the grammar code is repeated there, and three lines of annotation (telecodes, romanization, English gloss) are centered under the grammar code. Note that no provision is made for combining see-aboves and terminals; if a node is labeled with a reference number or contains a see-above line it is assumed to be non-terminal (this is not a restriction, but the result of a conscious decision).

Provision is made for one special case; if p1 is negative, no line is drawn, but a grammar code is written at the point specified by p3 and p4 (used for the top node of the tree).

Comments: Basic plotting parameters, such as character size, are inherited from the original call to TOMFUNG which initialized this plotting frame.

Name and Type: Subroutine SERIAL (an entry point to TOMFUNG) in overlay (5,4)

Function: writes a master serial number on each frame plotted

Called By: PLOTREE

Calls: TITLEG (a part of the proprietary Graphic Display System)

Reads: nothing

Writes: FORTRAN logical tape 99

Parameters: 1 - the serial number for this frame. May be either (a) an hollerith constant, or (b) an array containing the display code number--blank filled, last word left-justified and blank filled, first non-number word all octal zeroes.

General Description: SERIAL writes a serial number (or any other identifying information) at the lower right of the plotting frame.

SERIAL changes some of the general TOMFUNG parameters relating to character size, but restores them before returning.

Comments: Through the device of multiple entry points, SERIAL shares the communications areas of TOMFUNG and the other plotting routines.

Name and Type: Subroutine TOMFUNG in overlay (5,4)

Function: initialize a plotting frame for graphic display of SAS parse trees

Called By: PLOTREE

Calls: LLS (plus TITLEG, PLOT30, and SLLILI, which are part of the proprietary Graphic Display System)

Reads: FORTRAN logical tape number 40 (pre-selected Chinese character vectors)

Writes: FORTRAN logical tape number 99 (intermediate file)

Parameters: 1 - number of character positions along the X-axis (floating point)

2 - number of levels in tree along the Y-axis, including one level for terminals but no allowance for associated annotation of terminals (floating-point)

3 - returns 0.0 to indicate that no plotting is possible because the density of information is too great to be labeled

General Description: TOMFUNG first checks the number of character positions and levels against constant limits (currently 5000 characters and 95 levels); if either is exceeded, initialization is aborted and p3 returns -1.0. Future calls to plot routines after such an abort will be fielded and returned properly, but no output will be produced.

If the requested plot is within those limits, all housekeeping for plotting is carried out. Trees of less than fifteen levels are plotted on 11-inch paper; trees of fifteen levels or more generate a request for 29-inch paper. In either case the tree is spread out so that the plot occupies all space on the paper not needed for annotation. Character sizes are chosen from three sets of possibilities depending on the number of characters to be plotted.

TOMFUNG then reads the telecode text from COMMON/TEXT/, plots it along the bottom of the paper, and reads the character vectors from tape 41, plotting each character above its telecode; each telecode is also annotated with its sentence position.

Comments: The reading of vectors from tape 41 is a temporary arrangement; vectors will be stored in ECS when it is clear what plotter hardware will be standard, and thus what ECS packing scheme is optimal.

As a device to permit a shared communication pool, subroutines PLOT, NEWTREE, ENDPLOT, LABEL, and SERIAL--logically independent--are written as further entry points to this (formal) routine.

Name and Type: Subroutine DLLILI in overlay (5,4)

Function: Sets up plot calls for dashed-line drawing

Called by: PLOT

Calls: internal routines of the Graphic Display System

Reads: nothing

Writes: FORTRAN logical tape 99

Parameters: documented in GDS manual

General Description: [DLLILI is not part of the SAS system, but rather of the GDS System used for plotting.]

Comments: DLLILI is part of the proprietary Graphic Display System, and would normally be available only in object code; until it is made a part of the regular system library for GDS, it must be incorporated into user programs. As a courtesy to us, a source deck was made available to ease our problems in overlay handling. We have no internal documentation of this routine, nor any notion of the significance of its calls.

Name and Type: Subroutine DSHLNE in overlay (5,4)

Function: generates actual line segments for dashed-line drawing

Called By: DLLILI (a GDS routine)

Calls: internal routines of the Graphic Display System

Reads: nothing

Writes: FORTRAN logical tape 99

Parameters: documented in GDS manual

General Description: [DSHLNE is not part of the SAS system, but rather of the GDS System used for plotting]

Comments: DSHLNE is part of the proprietary Graphic Display System, and would normally be available only in object code; until it is incorporated in the regular system library for GDS, it must be included in user programs. As a courtesy to us, a source deck was made available to ease our problems in overlay handling. We have no internal documentation of this routine, nor any notion of the significance of its calls.

Name and Type: Subroutine PLOT30 in overlay (5,4)

Function: generates a plot-time request for hanging of 29-inch paper

Called By: TOMFUNG

Calls: internal routines of the Graphic Display System

Reads: nothing

Writes: FORTRAN logical tape 99

Parameters: documented in the GDS manual

General Description: [PLOT30 is not part of the SAS system, but rather of the GDS System used for plotting.]

Comments: PLOT30 is part of the proprietary GDS, and would normally be available only in object code; until it is made a part of the regular system library for GDS, it must be incorporated in user programs. As a courtesy to us, a source deck was made available to ease our problems in overlay handling. We have no internal documentation on this routine, nor any notion of the significance of its calls.

Name and Type: Subroutine SIGNON in overlay (5,4)

Function: initializes intermediate plotting files

Called By: internal routines of the Graphic Display System

Calls: internal routines of the Graphic Display System

Reads: nothing

Writes: FORTRAN logical tape 99

Parameters: [internal to GDS]

General Description: [SIGNON is not part of the SAS system, but rather of the GDS system used for plotting]

Comments: SIGNON is part of the proprietary Graphic Display System, and is normally available on the system GDS library only in object code. The system library version is not suitable for our purposes, since SIGNON (uniquely) depends on modifying a status word--and if that status word is located in a high-level overlay its status will be lost each time a new copy of the overlay is obtained.

As a courtesy to us, a source deck was made available to us and we have modified it by inserting a card declaring the status word as the contents of the one-word common block COMMON/PARADIS/, located in overlay 0,0. In this way the status is saved properly in spite of overlay swapping.

Apart from this custom modification, we have no internal documentation of this routine, nor any notion of the significance of its calls.

Name and Type: Subroutine TITLEG in overlay (5,4)

Function: Sets up plot calls for centered annotations

Called By: TOMFUNG, PLOT, SERIAL

Calls: internal routines of the Graphic Display System

Reads: nothing

Writes: FORTRAN logical tape 99

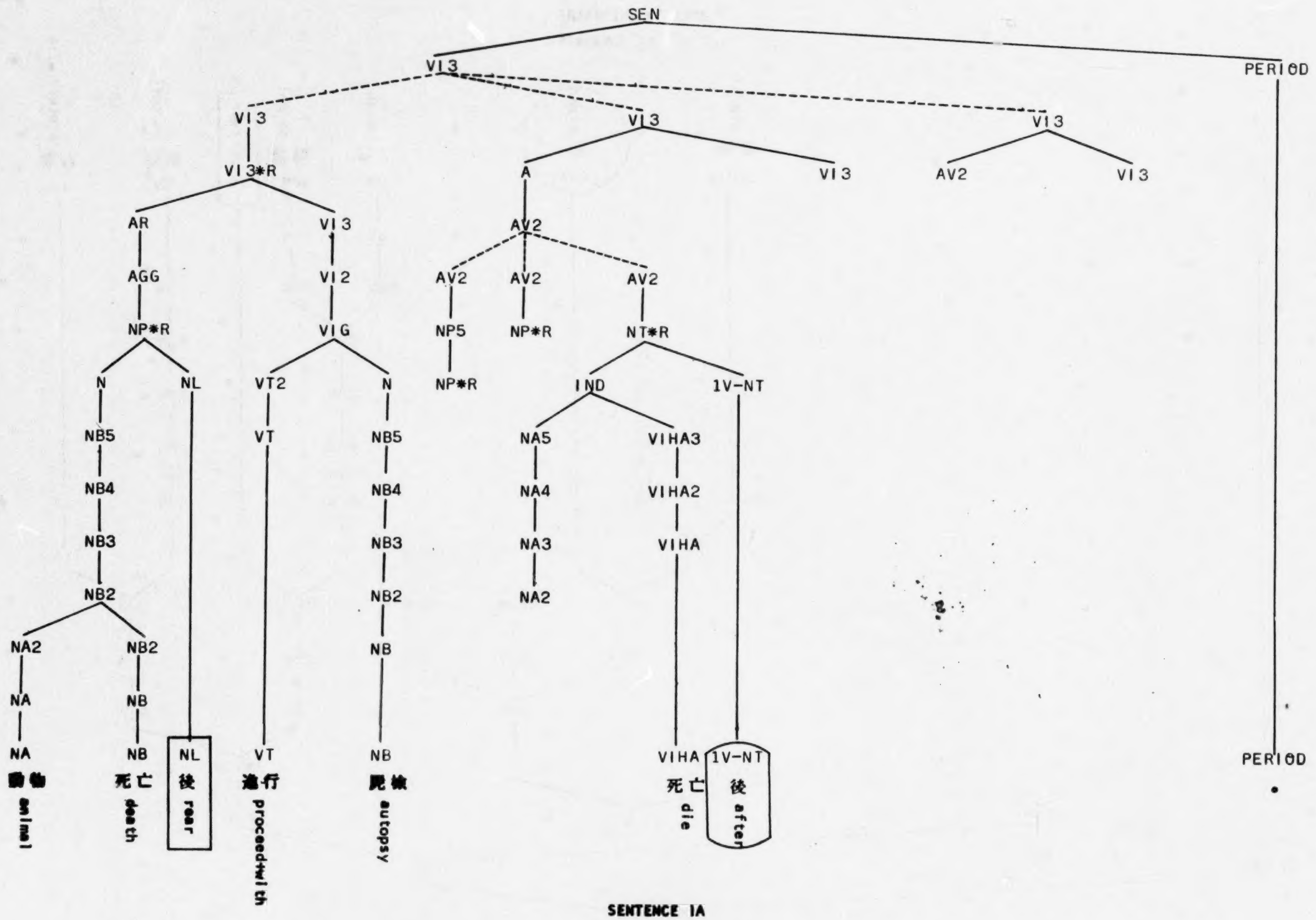
Parameters: documented in GDS manual

General Description: [TITLEG is not part of the SAS system, but rather of the GDS system used for plotting]

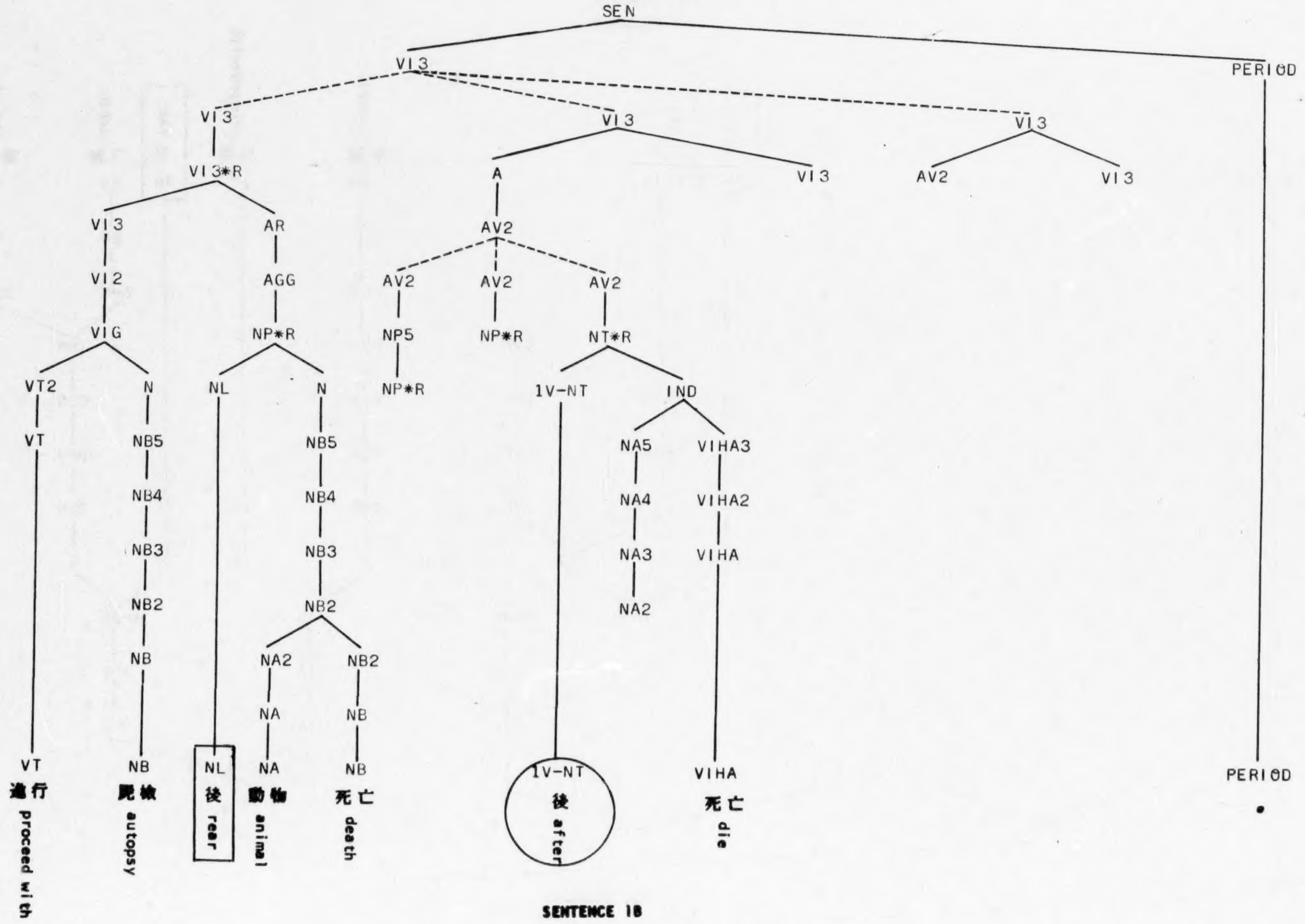
Comments: TITLEG is part of the proprietary Graphic Display System, and is normally available on the system plotting library only in object code. For some time a bug has existed in the system library version, and so we acquired a corrected copy to incorporate in our own programs. As a courtesy to us, a source deck was made available to ease our handling of overlays. We have no internal documentation on this routine, nor any notion of the significance of its calls.

APPENDIX III

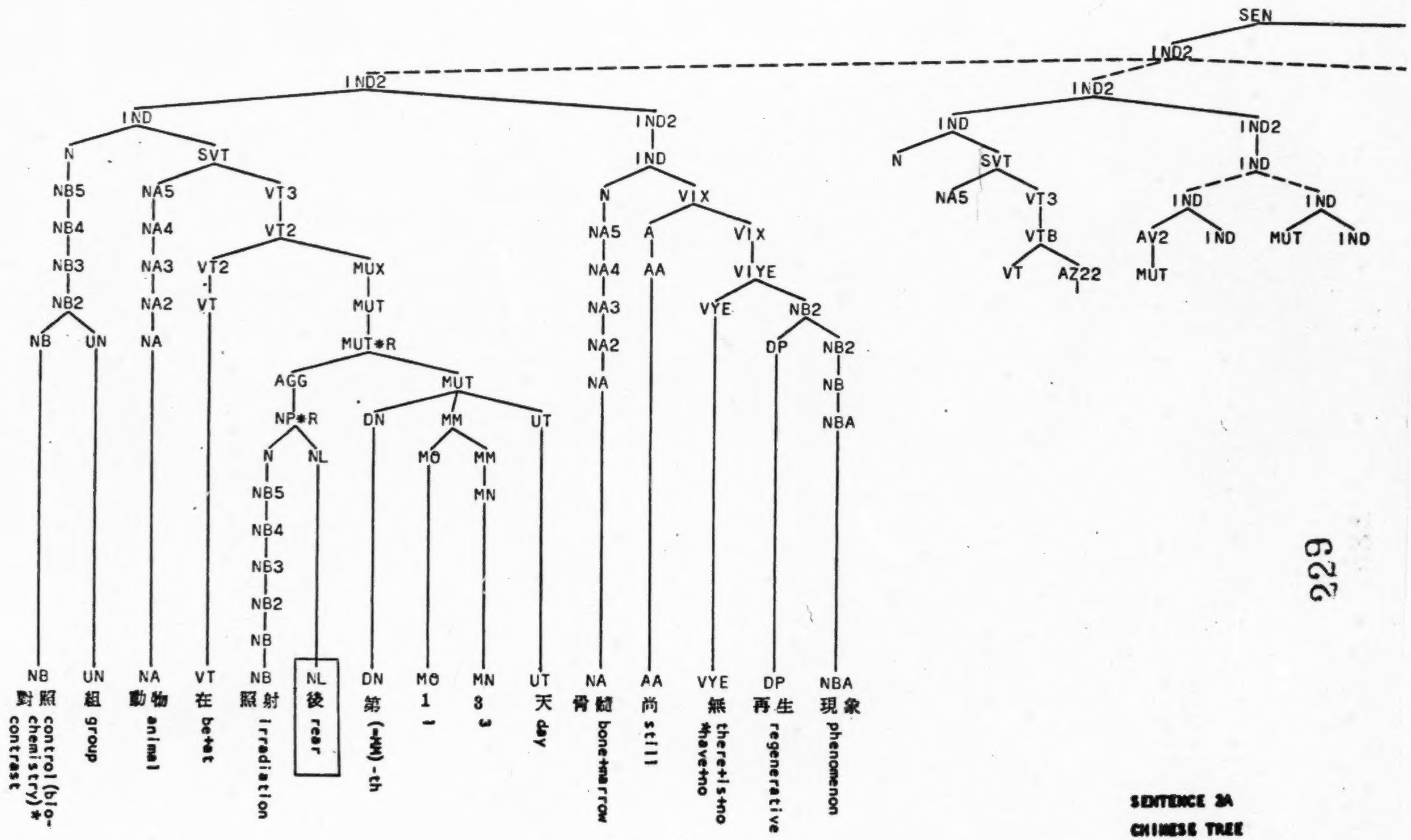
SAMPLES OF PLOTTED TREES


SEN
VI3
PERIOD
VI3
VI3
VI3
VI3*R
A
VI3
AV2
VI3
AR
VI3
AV2
AGG
VI2
AV2
AV2
AV2
NP*R
VIG
NP5
NP*R
NT*R
N
NL
VT2
N
NP*R
IND
1V-NT
NB5
VT
NB5
NA5
VIHA3
NB4
NB4
NA4
VIHA2
NB3
NB3
NA3
VIHA
NB2
NB2
NA2
NA2
NB2
NB
NA
NB
NA
NB
NL
VT
NB
VIHA
1V-NT
PERIOD
動物
animal
死亡
death
後
rear
進行
proceed+with
屍檢
autopsy
死亡
die
後
after
.

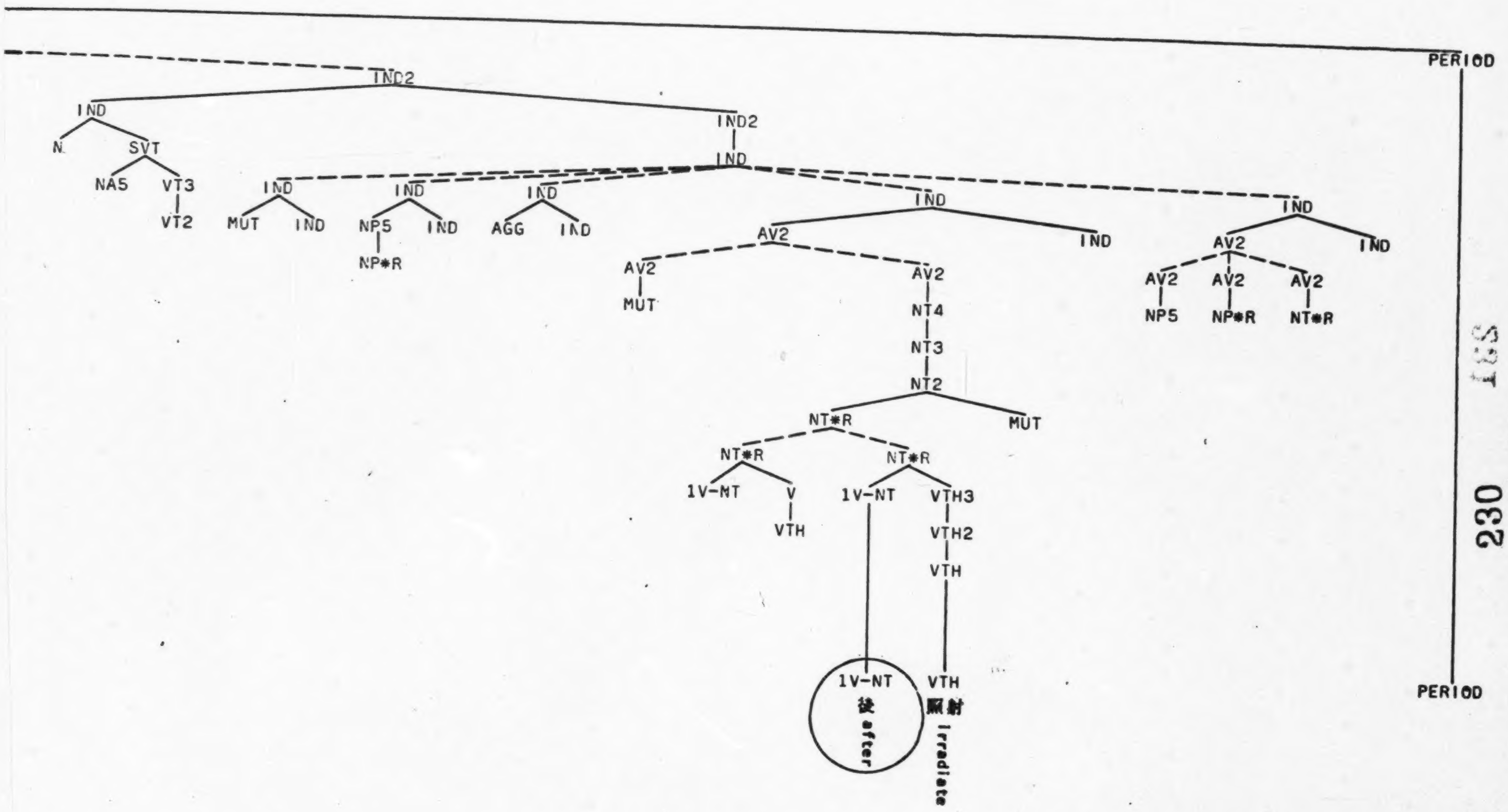
SENTENCE IA
CHINESE TREE


SEN
VI3
PERIOD
VI3
VI3*R
VI3
VI2
VIG
VT2
VT
N
NB5
NB4
NB3
NB2
NB
AR
AGG
NP*R
NL
N
NB5
NB4
NB3
NB2
NA2
NA
NB2
NB
VI3
A
AV2
AV2
NP5
NP*R
AV2
NP*R
AV2
NT*R
1V-NT
IND
NA5
NA4
NA3
NA2
VIHA3
VIHA2
VIHA
VI3
VI3
AV2
VI3
VT
進行
proceed with
NB
屍檢
autopsy
NL
後
rear
NA
動物
animal
NB
死亡
death
1V-NT
後
after
VIHA
死亡
die
PERIOD
.

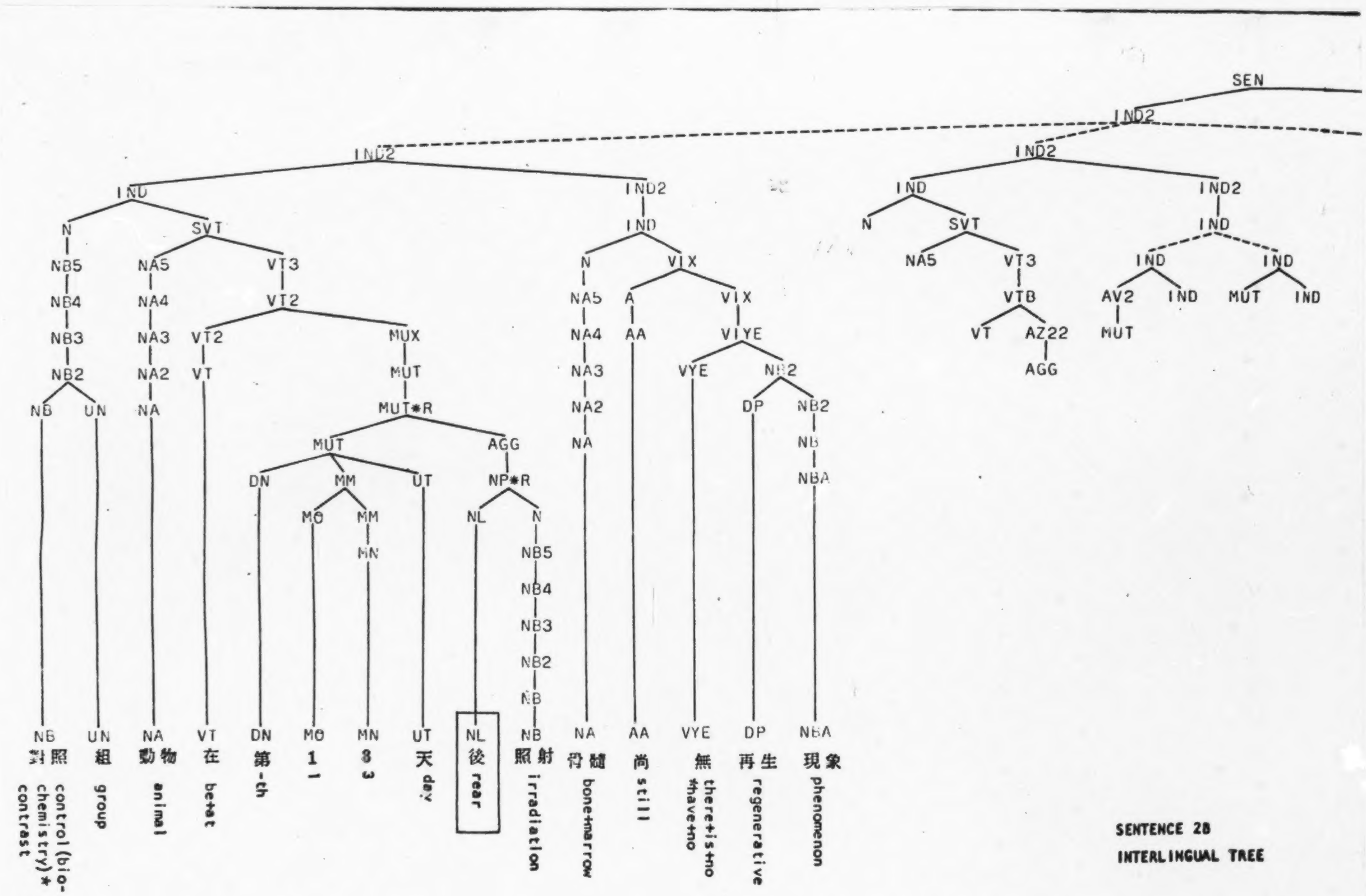
SENTENCE 18
INTERLINGUAL TREE


SEN
IND2
IND2
IND2
IND
N
SVT
NB5
NB4
NB3
NB2
NA5
NA4
NA3
NA2
VT3
VT2
VT2
VT
MUX
MUT
MUT*R
AGG
NP*R
N
NL
NB5
NB4
NB3
NB2
NB
MUT
DN
MM
MO
MM
MN
UT
IND2
IND
N
NA5
NA4
NA3
NA2
NA
VIX
A
AA
VIX
VIYE
VYE
NB2
DP
NB2
NB
NBA
IND
N
SVT
NA5
VT3
VTB
VT
AZ22
IND2
IND
IND
AV2
MUT
IND
IND
MUT
IND
NB 對照 control(bio-chemistry)* contrast
UN 組 group
NA 動物 animal
VT 在 be+at
NB 照射 irradiation
NL 後 rear
DN 第 (=MM)-th
MO 1 1
MN 3 3
UT 天 day
NA 骨髓 bone+marrow
AA 尚 still
VYE 無 there+is+no *have+no
DP 再生 regenerative
NBA 現象 phenomenon

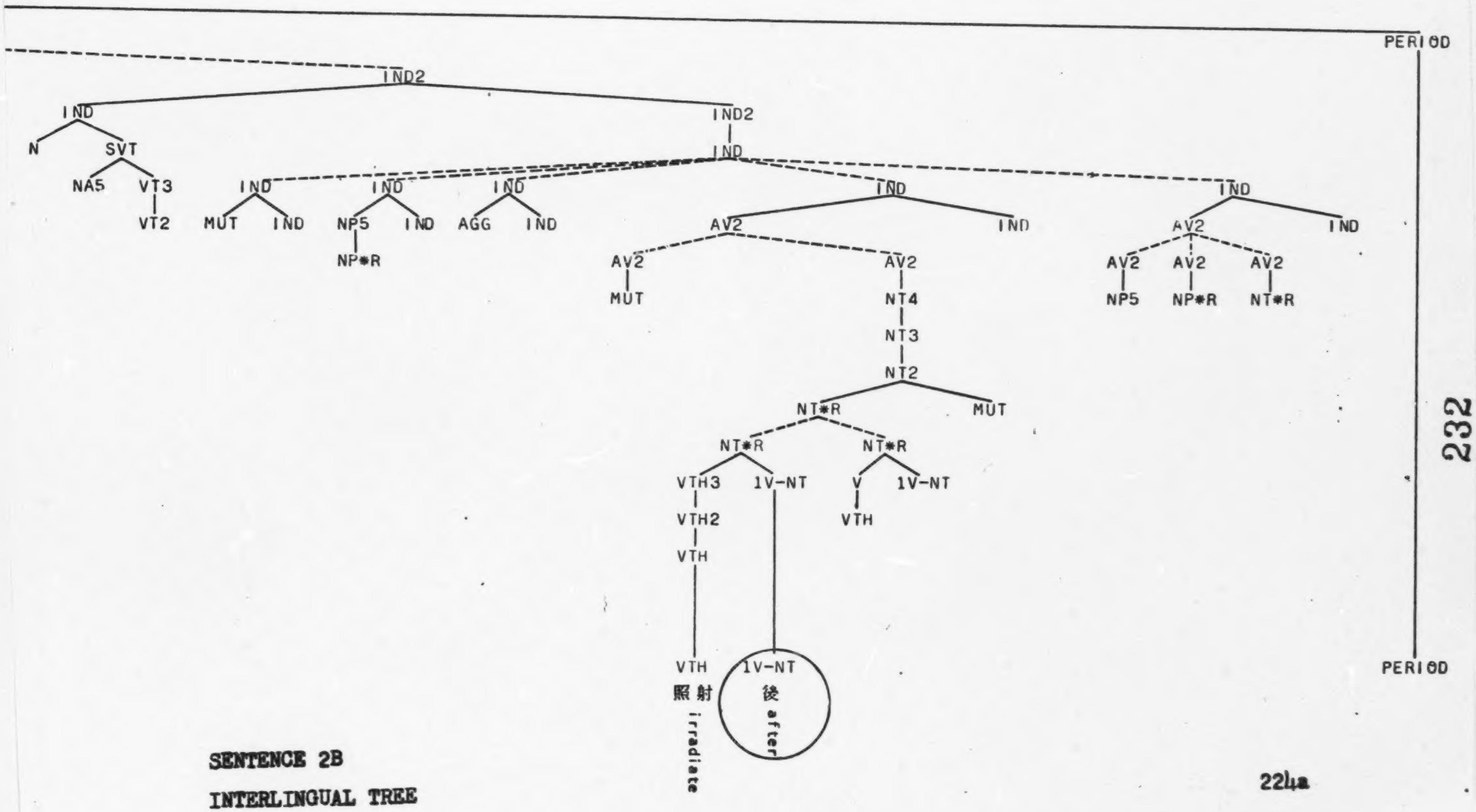
SENTENCE 3A
CHINESE TREE


PERIOD
IND2
IND
N
SVT
NA5
VT3
VT2
IND2
IND
IND
MUT
IND
IND
NP5
NP*R
IND
IND
AGG
IND
IND
AV2
AV2
MUT
AV2
NT4
NT3
NT2
NT*R
NT*R
1V-NT
V
VTH
NT*R
1V-NT
VTH3
VTH2
VTH
1V-NT
後 after
VTH
照射 irradiate
MUT
IND
IND
AV2
AV2
NP5
AV2
NP*R
AV2
NT*R
IND
PERIOD

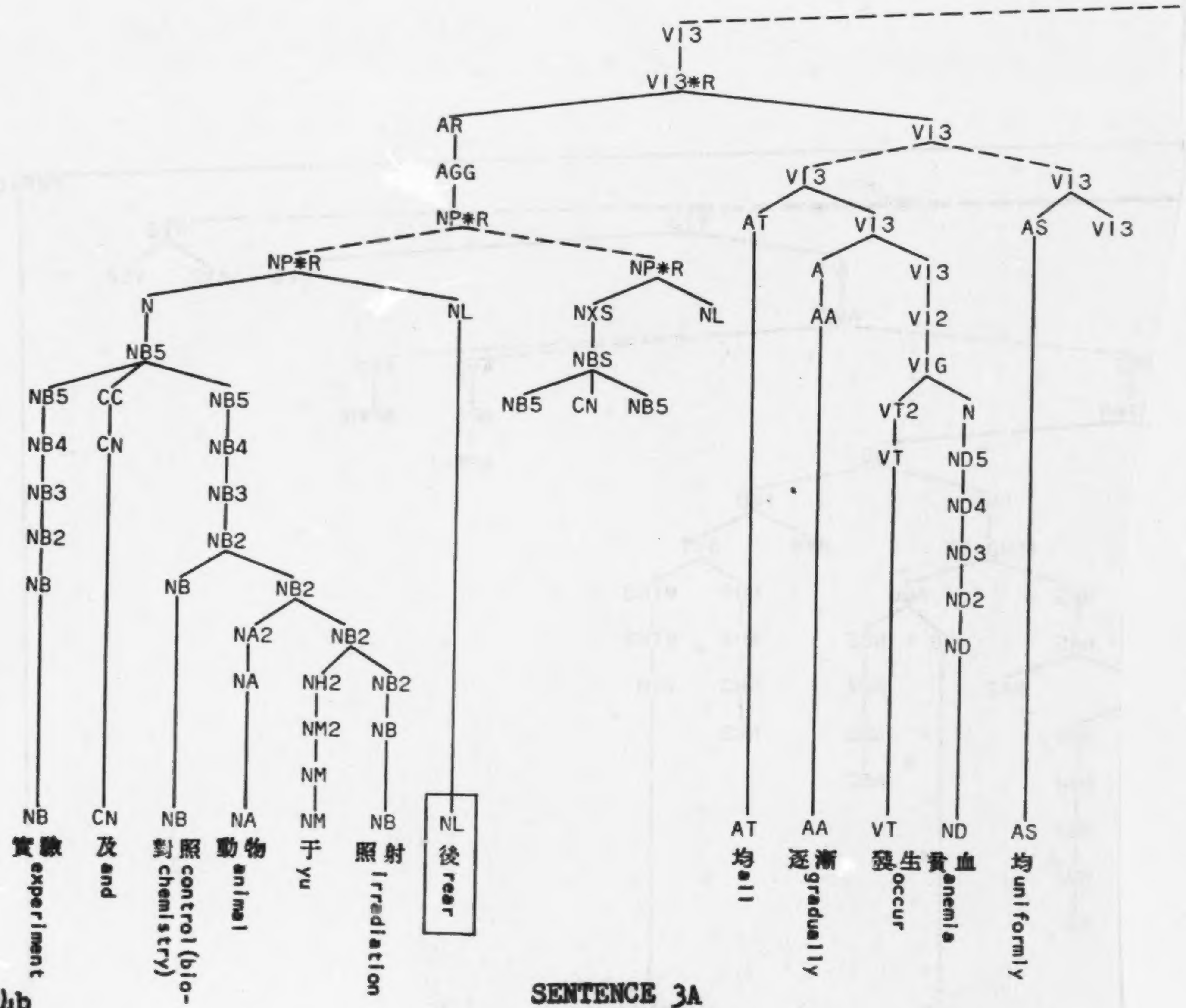
SENTENCE 2A
CHINESE TREE


SEN
IND2
IND
IND2
N
SVT
VIX
VIYE
VT3
VT2
MUX
MUT
MUT*R
AGG
NP*R
VTB
AZ22
AV2
NB 對照 control(bio-chemistry)* contrast
UN 組 group
NA 動物 animal
VT 在 be+at
DN 第 -th
M0 1 1
MN 8 3
UT 天 day
NL 後 rear
NB 照射 irradiation
NA 骨髓 bone+marrow
AA 尚 still
VYE 無 there+is+no *have+no
DP 再生 regenerative
NBA 現象 phenomenon

SENTENCE 28
INTERLINGUAL TREE

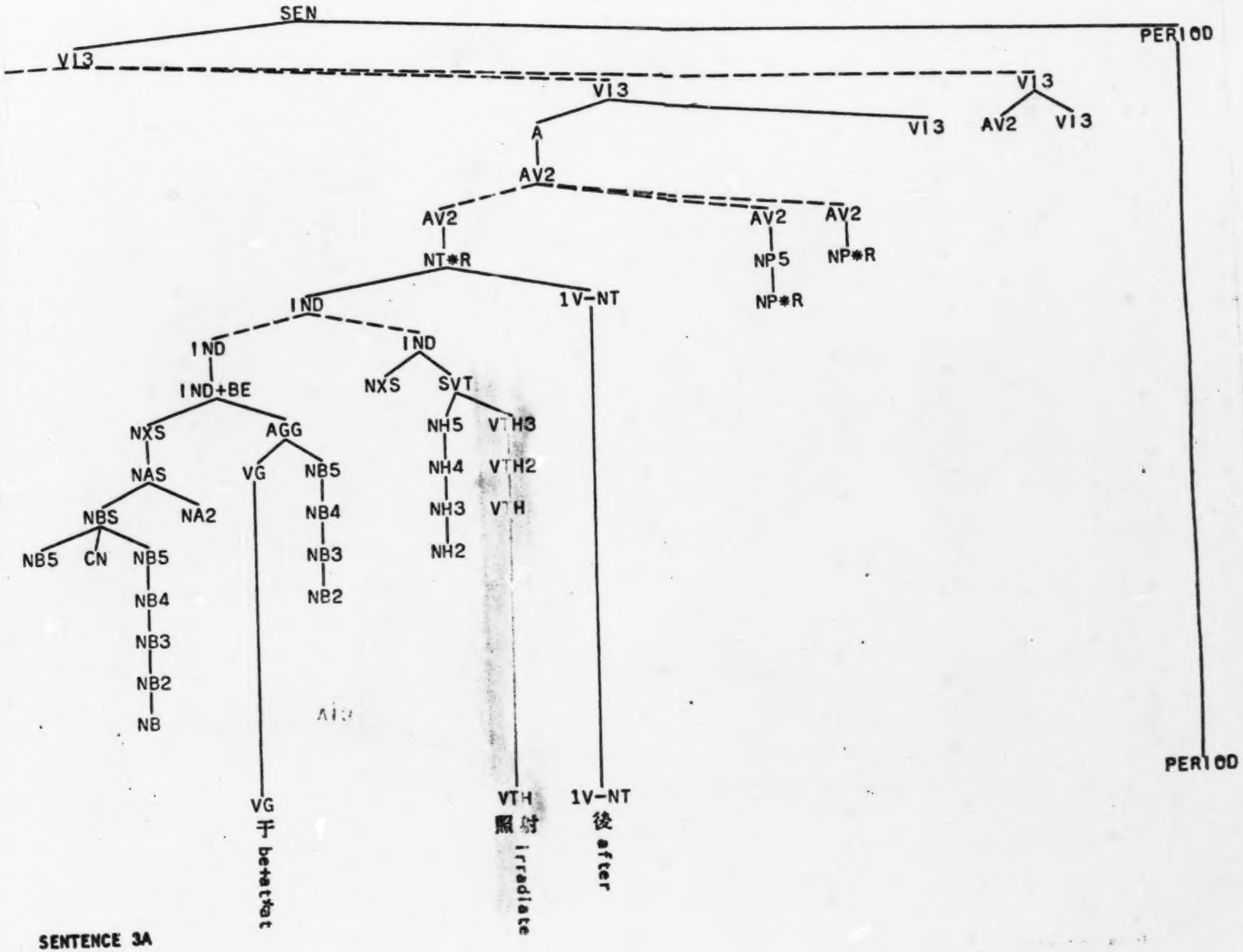

PERIOD
IND2
IND
N
SVT
NA5
VT3
VT2
IND2
IND
IND
MUT
IND
IND
NP5
IND
NP*R
IND
AGG
IND
IND
AV2
IND
AV2
MUT
AV2
NT4
NT3
NT2
NT*R
MUT
NT*R
NT*R
VTH3
1V-NT
V
1V-NT
VTH2
VTH
VTH
VTH
1V-NT
照射
irradiate
後
after
IND
AV2
IND
AV2
NP5
AV2
NP*R
AV2
NT*R
PERIOD

SENTENCE 2B
INTERLINGUAL TREE

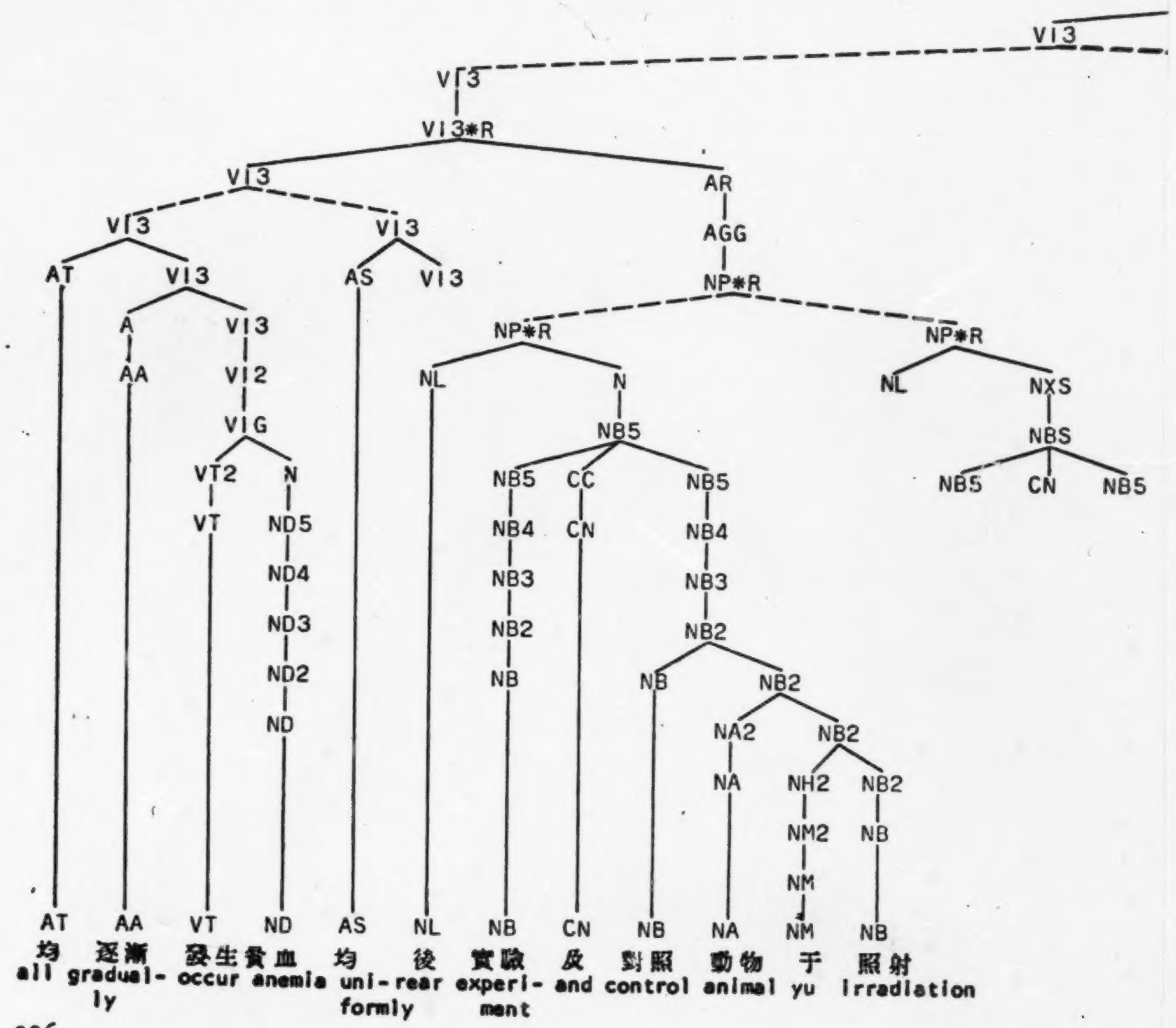

VI3
VI3*R
AR
AGG
NP*R
NP*R
N
NB5
NB5
NB4
NB3
NB2
NB
CC
CN
NB5
NB4
NB3
NB2
NB
NB2
NA2
NA
NB2
NH2
NM2
NM
NB2
NB
NL
NP*R
NXS
NBS
NB5
CN
NB5
NL
VI3
VI3
AT
VI3
A
AA
VI3
VI2
VIG
VT2
VT
N
ND5
ND4
ND3
ND2
ND
VI3
AS
VI3
NB
實驗 experiment
CN
及 and
NB
對照 control (bio-chemistry)
NA
動物 animal
NM
于 yu
NB
照射 irradiation
NL
後 rear
AT
均 all
AA
逐漸 gradually
VT
發生 occur
ND
貧血 anemia
AS
均 uniformly

224b

SENTENCE 3A
CHINESE TREE


SEN
VI3
PERIOD
VI3
VI3
VI3
AV2
VI3
A
AV2
AV2
AV2
AV2
NT*R
NP5
NP*R
NP*R
IND
1V-NT
IND
IND
IND+BE
NXS
SVT
NXS
AGG
NH5
VTH3
NAS
VG
NB5
NH4
VTH2
NBS
NA2
NB4
NH3
VTH
NB5
CN
NB5
NB3
NH2
NB4
NB2
NB3
NB2
NB
PERIOD
VG
于
be+at*at
VTH
照射
irradiate
1V-NT
後
after

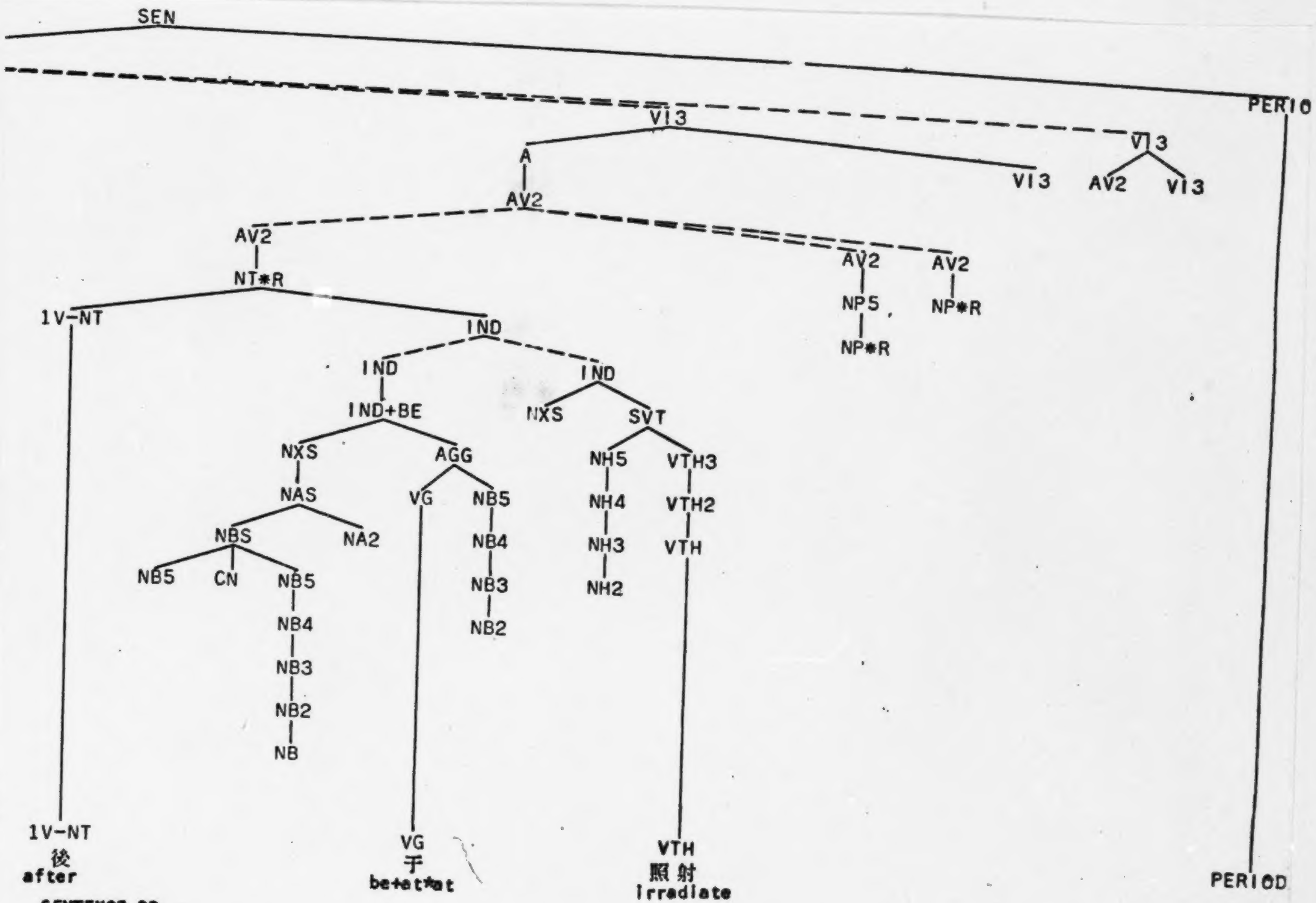
SENTENCE 3A
CHINESE TREE



VI3
VI3
VI3*R
VI3
AR
VI3
VI3
AGG
AT
VI3
AS
VI3
NP*R
A
VI3
NP*R
NP*R
AA
VI2
NL
N
NL
NXS
VIG
NB5
NBS
VT2
N
NB5
CC
NB5
NB5
CN
NB5
VT
ND5
NB4
CN
NB4
ND4
NB3
NB3
ND3
NB2
NB2
ND2
NB
NB
NB2
ND
NA2
NB2
NA
NH2
NB2
NM2
NB
NM
AT
AA
VT
ND
AS
NL
NB
CN
NB
NA
NM
NB
均
逐漸
發生
貧血
均
後
實驗
及
對照
動物
于
照射
all
gradual-ly
occur
anemia
uni-formly
rear
experi-ment
and
control
animal
yu
irradiation

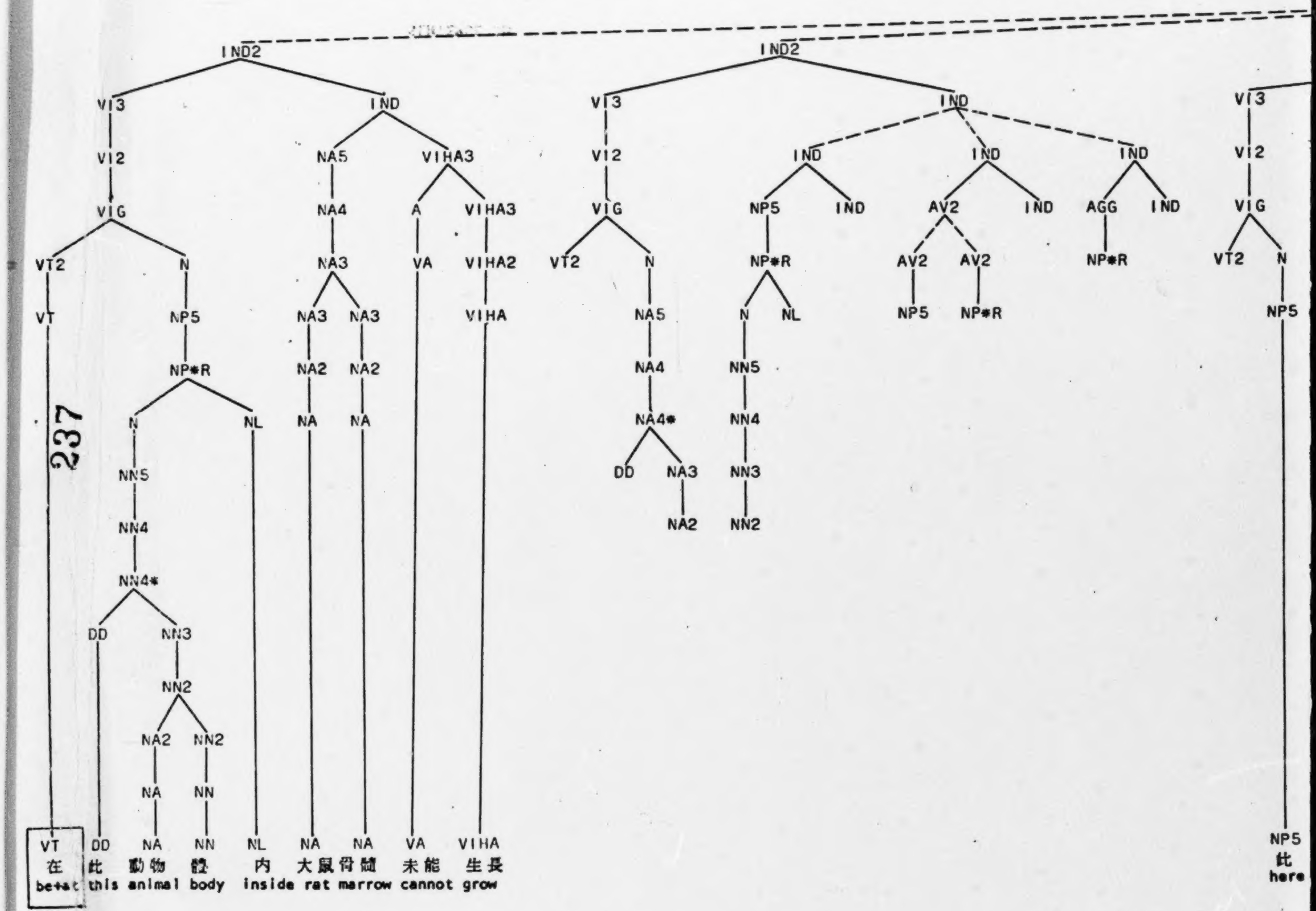
SENTENCE 3B

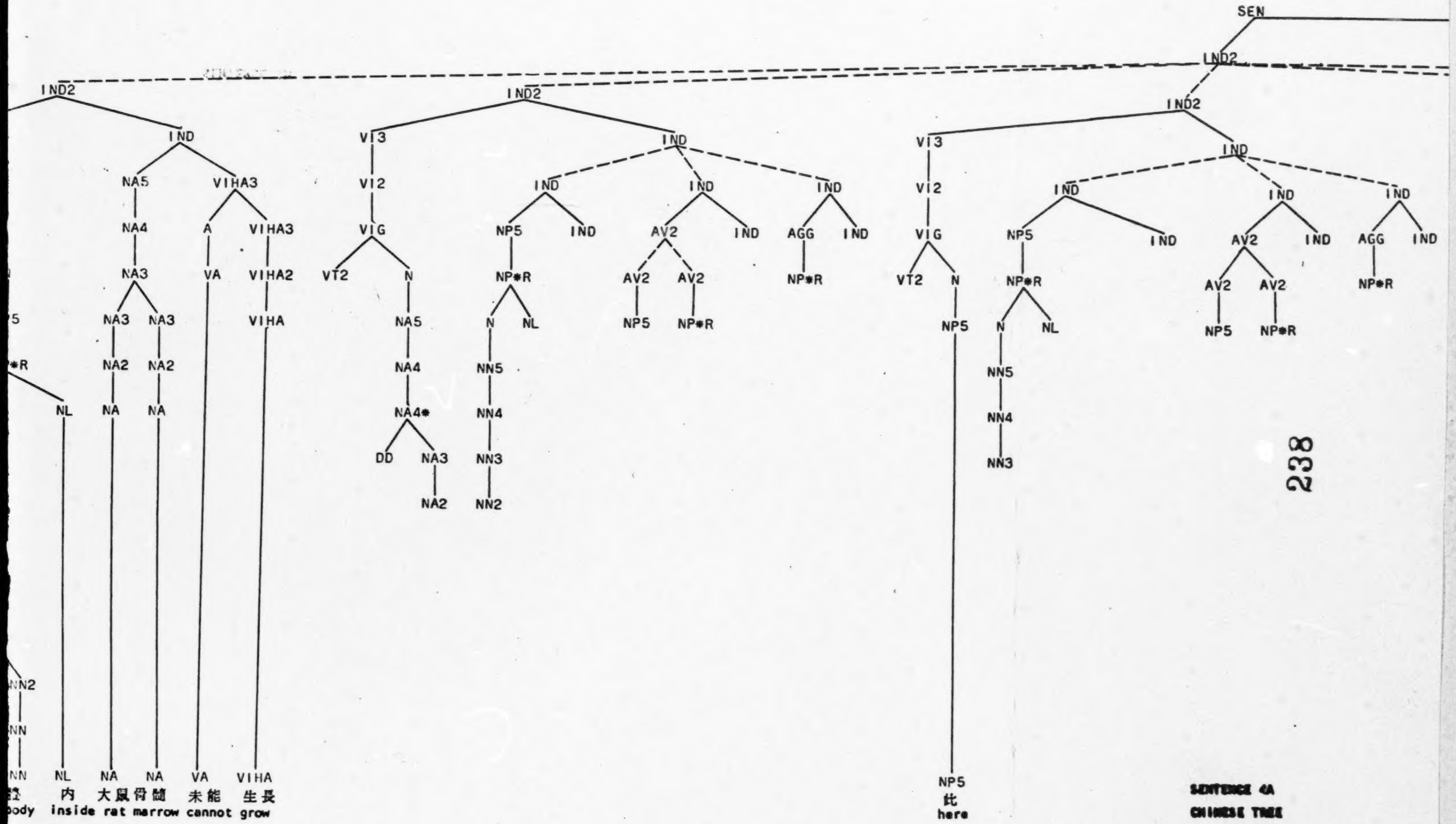
INTERLINGUAL TREE


SEN
PERIO
VI3
A
AV2
VI3
VI3
AV2
VI3
AV2
NT*R
AV2
NP5 ,
NP*R
AV2
NP*R
1V-NT
IND
IND
IND+BE
IND
NXS
SVT
NXS
NAS
NBS
NA2
NB5
CN
NB5
NB4
NB3
NB2
NB
AGG
VG
NB5
NB4
NB3
NB2
NH5
NH4
NH3
NH2
VTH3
VTH2
VTH
1V-NT
後
after
VG
于
be+at*at
VTH
照射
irradiate
PERIOD

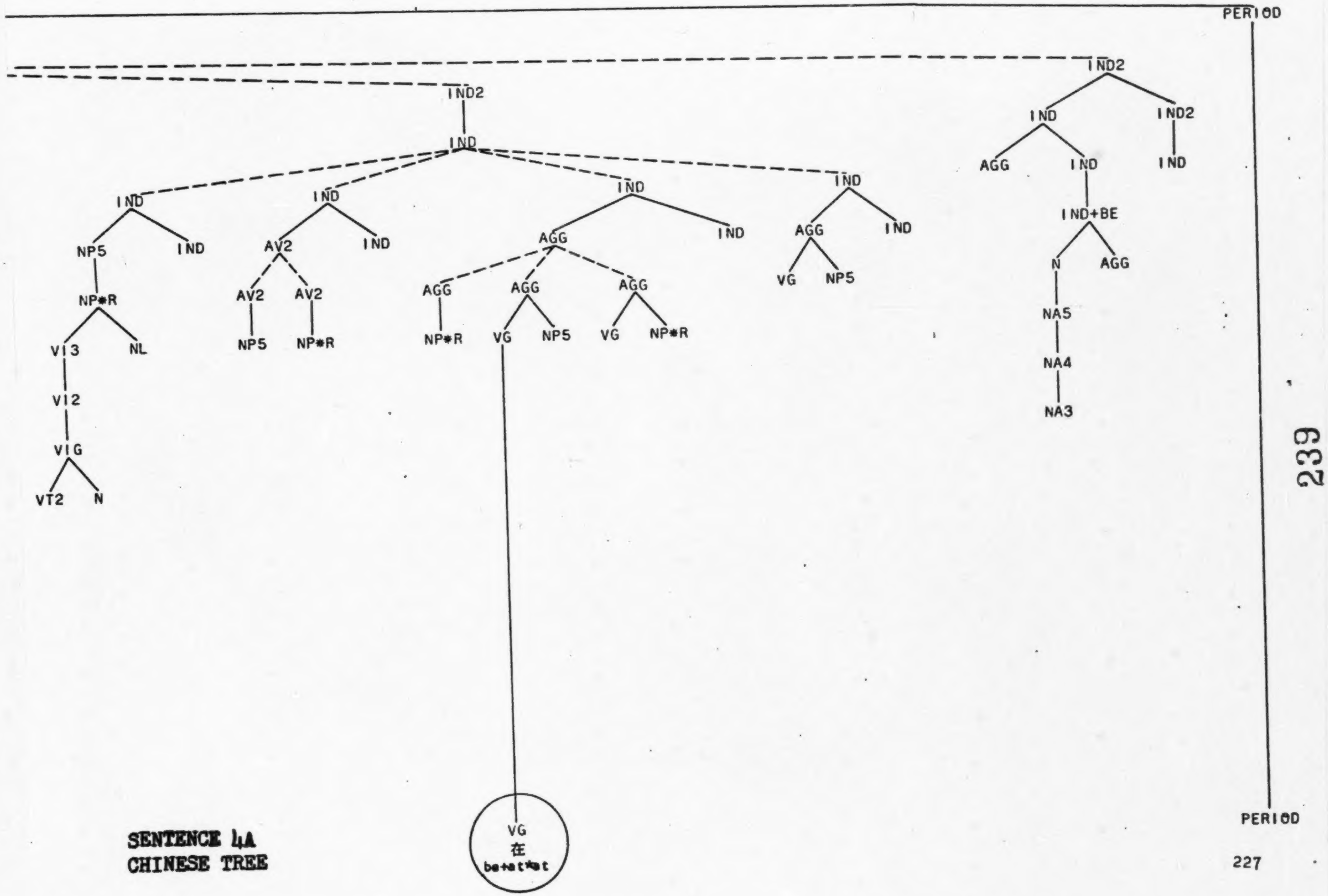
SENTENCE 3B
INTERLINGUAL TREE

226a

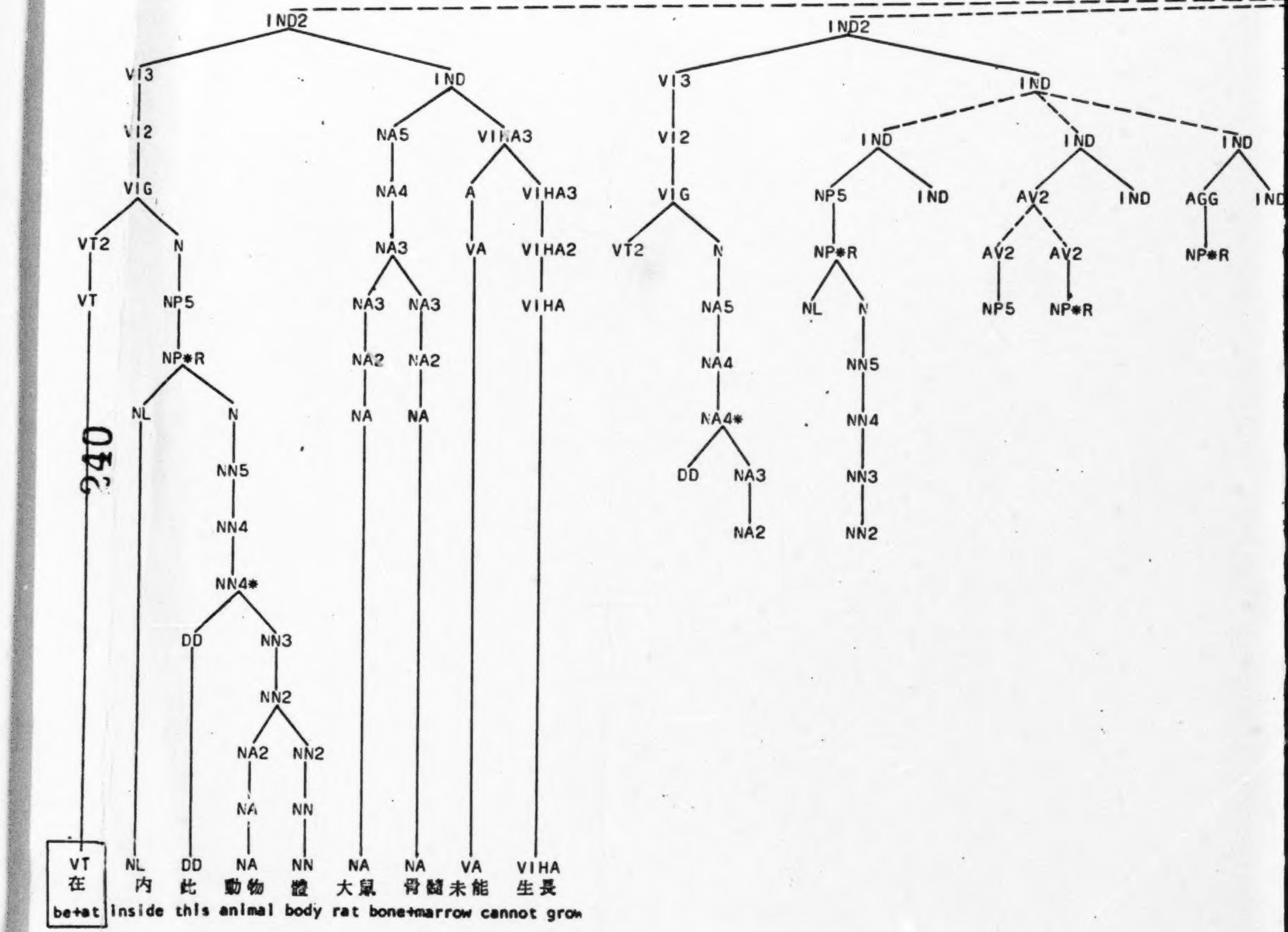

IND2
VI3
IND
VI2
NA5
VIHA3
VIG
NA4
A
VIHA3
VT2
N
NA3
VA
VIHA2
VT
NP5
NA3
NA3
VIHA
NP*R
NA2
NA2
N
NL
NA
NA
NN5
NN4
NN4*
DD
NN3
NN2
NA2
NN2
NA
NN
VT
在
be+at
DD
此
this
NA
動物
animal
NN
體
body
NL
內
inside
NA
大鼠
rat
NA
骨髓
marrow
VA
未能
cannot
VIHA
生長
grow
IND2
VI3
IND
VI2
IND
IND
IND
VIG
NP5
IND
AV2
IND
AGG
IND
VT2
N
NP*R
AV2
AV2
NP*R
NA5
N
NL
NP5
NP*R
NA4
NN5
NA4*
NN4
DD
NA3
NN3
NA2
NN2
VI3
VI2
VIG
VT2
N
NP5
NP5
此
here

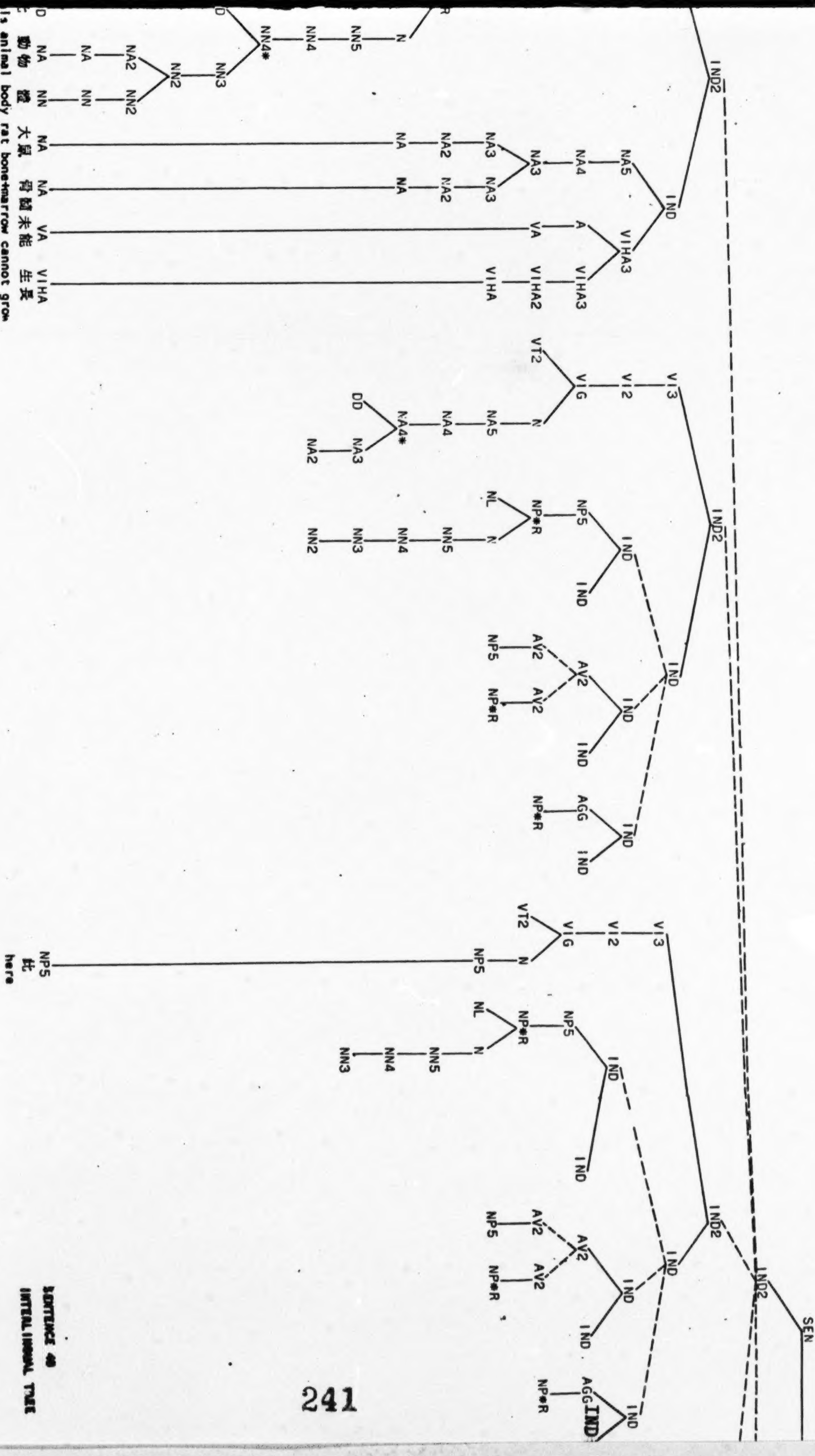

SEN
IND2
IND2
IND2
IND2
IND
NA5
NA4
NA3
NA3
NA3
NA2
NA2
NA
NA
VIHA3
A
VA
VIHA3
VIHA2
VIHA
NP*R
NL
NN2
NN
NN
NL
NA
NA
VA
VIHA
體
內
大鼠
骨髓
未能
生長
body
inside
rat
marrow
cannot
grow
VI3
VI2
VIG
VT2
N
NA5
NA4
NA4*
DD
NA3
NA2
IND
IND
NP5
IND
NP*R
N
NL
NN5
NN4
NN3
NN2
IND
AV2
IND
AV2
AV2
NP5
NP*R
IND
AGG
IND
NP*R
VI3
VI2
VIG
VT2
N
NP5
NP5
此
here
IND
IND
NP5
IND
NP*R
N
NL
NN5
NN4
NN3
IND
AV2
IND
AV2
AV2
NP5
NP*R
IND
AGG
IND
NP*R

SENTENCE 4A
CHINESE TREE

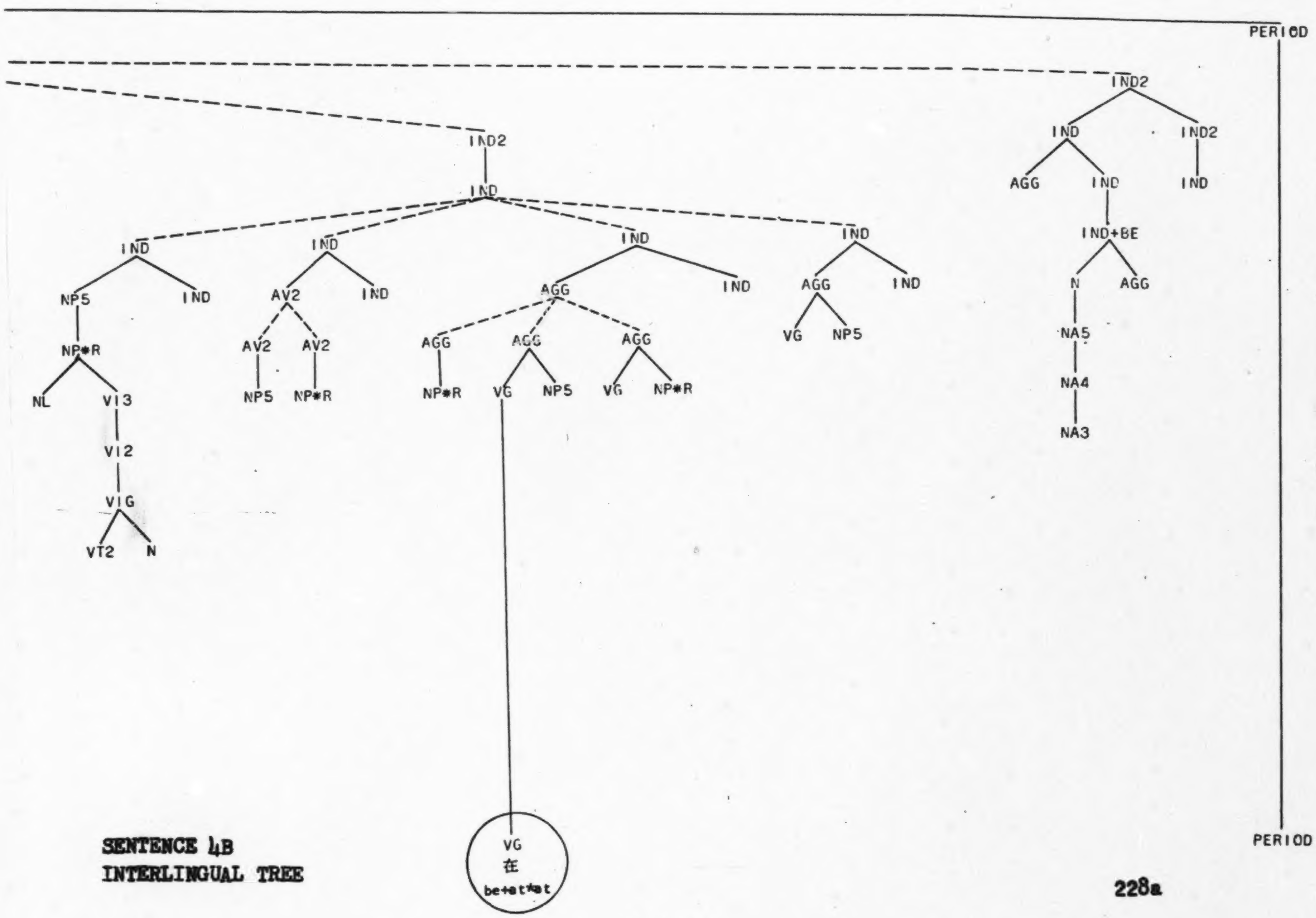

PERIOD
IND2
IND2
IND
IND
NP5
NP*R
VI3
VI2
VIG
VT2
N
NL
IND
IND
AV2
AV2
NP5
AV2
NP*R
IND
IND
AGG
AGG
NP*R
AGG
VG
NP5
AGG
VG
NP*R
IND
IND
AGG
VG
NP5
IND
IND
AGG
IND
IND+BE
N
NA5
NA4
NA3
AGG
IND2
IND
VG
在
be+at*at
PERIOD

SENTENCE 4A
CHINESE TREE


IND2
VI3
VI2
VIG
VT2
VT
N
NP5
NP*R
NL
NN5
NN4
NN4*
DD
NN3
NN2
NA2
NA
IND
NA5
NA4
NA3
VIHA3
A
VA
VIHA2
VIHA
AV2
NP5
AGG
VT 在 be+at
NL 內 inside
DD 此 this
NA 動物 animal
NN 體 body
NA 大鼠 rat
NA 骨髓 bone+marrow
VA 未能 cannot
VIHA 生長 grow


SEN
IND2
IND2
IND2
IND
VI3
VI2
VIG
VT2
N
NP5
NP5
此
here
NP5
NP*R
NL
N
NN5
NN4
NN3
AV2
AV2
AV2
NP5
NP*R
AGG
NP*R
VIHA3
VA
A
VIHA
VIHA2
NA5
NA4
NA3
NA2
NA
NN3
NN2
NA2
NN2
NA4*
DD
NA
動物
NN
體
NA
大鼠
NA
骨髓
VA
未能
VIHA
生長
Is animal body rat bone-marrow cannot grow

SENTENCE 40
INTERLINGUAL TREE


PERIOD
IND2
IND
IND2
AGG
IND
IND
IND+BE
N
AGG
NA5
NA4
NA3
IND2
IND
IND
NP5
IND
NP*R
NL
VI3
VI2
VIG
VT2
N
IND
AV2
IND
AV2
AV2
NP5
NP*R
IND
AGG
IND
AGG
AGG
AGG
NP*R
VG
NP5
VG
NP*R
IND
AGG
IND
VG
NP5
VG
在
be+at*at
PERIOD

SENTENCE 4B
INTERLINGUAL TREE

Appendix IV. Chicoder Coding (or Decoding)

The Chicoder is a device for punching a tape representation of English or Chinese text entered through the keyboard. In English mode the Chicoder acts as an off-line teletype, producing one punch character for each key struck. This includes non-printing keys, such as tape keys, and the English and Chinese mode keys themselves. The operation of the Chicoder is, however, rather more complex in Chinese mode. In this mode each printing key also represents a top and bottom portion of (usually) several Chinese characters. When two of these keys are struck, a five-by-five grid is displayed showing the possible characters corresponding to these keys. Sequence and position keys indicate respectively the row and column in the grid of the desired character. If the character is not displayed, the machine can get reset and a new initial pair of keys can be tried. If the character is found, a code is punched when the position key is struck. The code consists of four punches, the first two being the usual punches for the printing keys struck, the third punch giving the binary representation of the row and column numbers, (each from 1 through 5), and the fourth a special on-off-character punch -- the same for all characters coded.

Using both the English and Chinese mode features, an operator can encode Chinese character text on tape, entering English mode to insert telecodes for punctuation of Chinese characters not represented on the Chicoder. Since the Chicoder neither punches characters in telecode representation, nor punches alpha-numeric characters in a standard teletype code, a program is being written to both translate and transliterate Chicoder code to telecodes. In English mode the transliteration of Chicoder mode to the appropriate internal character codes of the CDC 6400 simply requires a table of correspondences between the numerical codes assigned to each character. (This process is complicated slightly by the fact that, since the Chicoder punches are non-standard, the 8 bits of each punch position are broken up by the remote terminal system into two 6 bit internal characters. The 6 bits of the 8 which actually carry information have to be reconstituted as a single character before a correspondence can be determined.)

In Chinese mode, however, the transliteration is somewhat more cumbersome. In essence, most of the internal logical structure of the Chicoder must be duplicated in the translating program. The program will use the first two punches to choose a particular 5 by 5 matrix and then interpret the third punch, containing the row and column information, to select a cell wherein will be stored the telecode corresponding to the character punched. The fourth punch will also be examined to make sure that it is the end of character punch, no error has occurred in the punching. In addition other error checking will be done. For example, checking that the row and column indicated give possible positions within the grid.

Appendix V. The Sentence Generation Program (SGP)

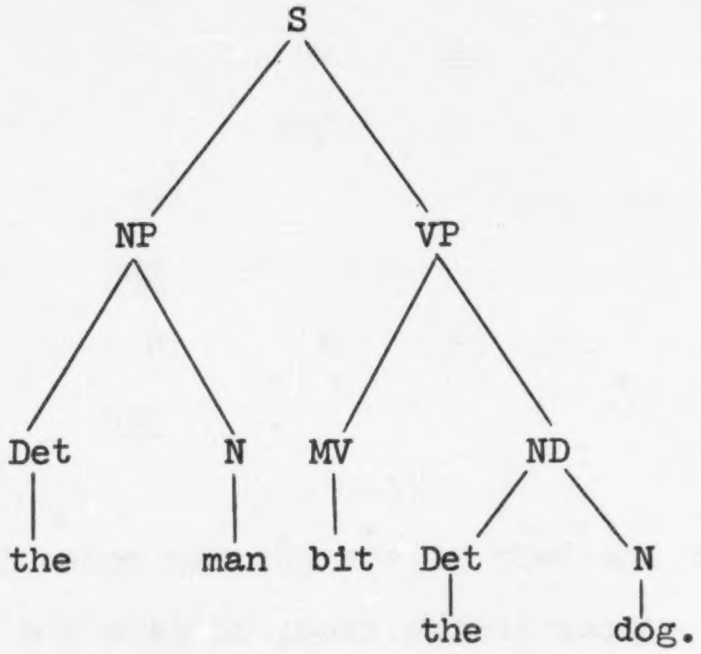
GENERAL: This program takes a grammar code and generates example expansions of it from a grammar. Grammars to be used with it have two parts, a set of phrase structures, and a dictionary of lexical entries. It operates with the most recent set of rules and a recent subdictionary of CHIDIC. There is a provision for adding new phrase structure rules in order to anticipate their effect on the grammar as a whole.

Subject to several constraints which insure finite termination of generation, it works as follows. The input for the SGP is (1) a set of new rules and (2) a set of generation instructions. The new rules are adapted into an efficient computer internal form and their use is integrated with the current rules. The generation instructions each contain a grammar code, parameters relating to generation constraints, and a parameter giving the number of generation-instances to be produced. Generation proceeds in pre-order -- that is the left-most daughter node of a given node is expanded before its sisters are. At any node (i.e. grammar code) two courses of generation are a priori possible: rule expansion and dictionary or lexical expansion. A choice is made randomly when both are actually possible, i.e. when there are both rules and lexical entries expanding the grammar code. Once this is decided a particular rule or lexical entry is selected randomly from those available and expansion is performed.

In order to make the addition of new phrase structure rules useful, expansion is mandatory on them. That is, when a grammar code is encountered for which there are new rule(s), such a rule must be selected for expansion.

All output is printed in a labelled pre-order format concurrent with generation. An example is in order:

Suppose the phrase-structure rules S → NP + VP, NP → Det + N, VP → MV + NP and the lexical entries Det → the, N → dog, N → Man, MV → bit, are selected during the course of generation so as to form the tree



The SGP program will output this expansion of the grammar code S in the form

```
1    S
2    NP
3    Det
               Det
               the
3    N
               N
               man
2    VP
3    MV
               MV
               bit
3    NP
4    Det
               Det
               the
4    N         N
               dog
```

The numbers to the left of the grammar codes indicate the level at which they appear in the tree; compare the output with the tree.

Aside from the basic input, viz., the new phrase structure rules and the generation instructions, the SGP requires several other portions of data; the program itself, the current rules, the current dictionary and dictionaries. When operated in pure batch mode the program is loaded on cards and the remaining ancillary information is loaded from tape. When operated from a teletype, all information is loaded from tape. In either case use is made of the disc files and the Extended Core Storage (ECS) facilities of the CDC 6400 for storing the large files necessary.

There now follows a more detailed explanation of the workings of the SGP. Since it is not highly modular this presentation simply follows the main flow of control.

INPUT: There are four blocks of information (in addition to the program itself) required for a generation run: two tape files and two sets of card (or teletype) input. The two tapes are referred to by their SCOPE (the CDC operating system) file names: TAPE1 and TAPE5. These particular names are a consequence of the FORTRAN file reading conventions.

(1) TAPE1 contains a dictionary sorted in ascending order on the BCD representations of the grammar codes of the entries.

(2) TAPE5 contains (a) the BCD representations of the grammar codes, (b) a table for referring to all current rules and dictionary entries for each grammar code, and (c) an adapted version of the current grammar rules. The rules are sorted on the BCD representation of the grammar codes they expand. The two other sets of input information are

(3) the new phrase structure rules -- the rules whose expansion will be mandatory whenever usable, and,

(4) generation instructions; these have four parts:

(a) the *grammar code* to be expanded

(b) a *level* beyond which generation by dictionary entry is forcing wherever possible.

(c) a *count parameter* which forces generation by dictionary entry whenever the number of occurrences of any grammar code exceeds the specified amount, and,

(d) the *number of generation instances* to be produced.

* * * * * *

The SGP is written in CDC Fortran and uses several utility routines.

SUBROUTINES: The utility routines perform such tasks as character handling, storage and retrieval from Extended Core Storage, and sorting. They are

(1) *SHIFTR*, a COMPASS (assembly language) routine for shifting the contents of a CDC 6400 word left or right.

(2) *DSORT*, and its subroutines NARGS, MXSWAP, KEQF; this is a package for performing a shell sort and is documented in the SAS writeup.

(3) RANDMOD, given an input value n it produces a (pseudo-) random integer between o and n-1. It is documented in the SAS writeup.

(4) ECS, this is a COMPASS (assembly language) program used for reading and writing segments of information from and to Extended Core Storage. It is documented in the SAS write-up.

A final subroutine named EXPAND is peculiar to the SGP; its operation is explained in the detailed explanation of the workings of the SGP (which is named GENUN).

THE WORKINGS OF THE SGP: The following is a description which parallels a printout of the program.

Initially a variety of arrays of variables, masks, and format statements are set. The information from file TAPE5 is then read into main core.

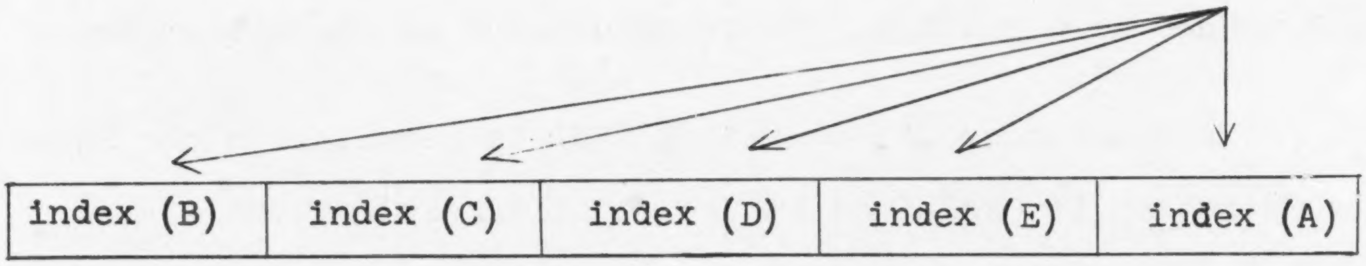
The dictionary information on TAPE1 is read into core and is transferred by segments into Extended Core Storage. All references to grammar codes except on output are by index. The index of a grammar code is its position in the grammar code table on TAPE5. (See above.)

After reading all the tape-stored information, the new rules are read into core. They are sorted and a list of grammar codes is prepared, grammar codes which are expanded by the new rules.

Using the index representation of grammar codes, new rules are packed into single CDC 6400 words. The grammar rule

$$A \rightarrow B + C + D + E$$

is represented in the computer by a word of the form:



12 bit segments in the 60 bit 6400 word

(This is exactly the same representation of rules which is used for the current grammar rules which were read from TAPE5.)

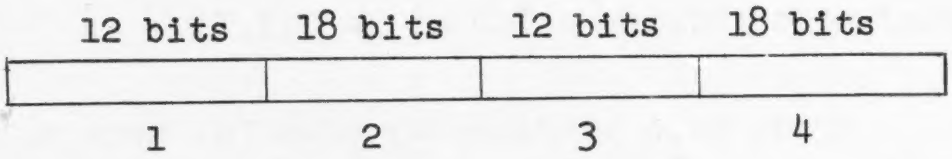
It would be useful at this point to explain the contents of the main grammar code table, labelled "G" in the program. There is one entry for each grammar code and it is composed of three 6400 words: G(1,I), G(2,I), G(3,I) for the Ith grammar code in the table.

G(1,I) contains the BCD representation of the Ith grammar code.

G(2,I) contains directory information for accessing the rules and dictionary entries expanding the Ith code.

It is of the following form:

G(2,I):



Portion 1 contains the number of dictionary entries.

Portion 2 contains the ECS address for the first dictionary entry expanding the Ith grammar code.

Portion 3 contains the number of rules.

Portion 4 contains the address in array R̲ of the first rule expanding the Ith grammar code.

G(3,I) is used to flag existing grammar codes used in new rules.

Since both the entries in dictionary and the rules in array R̲ are sorted by the grammar code, this is all the information needed for access.

After new rules containing unexpandable grammar codes (i.e., grammar codes for which there is no rule or dictionary expansion) have been eliminated, a reference table for grammar codes not appearing in G̲ is prepared, but only Portions 3 and 4 above are inserted here, since for these grammar codes no dictionary expansion will be possible.

Furthermore, all of the current grammar codes for which there are new rules are flagged. This is accomplished by placing the address of the entry for that grammar code in GNR (the new-grammar-code reference table) in G(3,I).

After this preparatory work has been accomplished, generation instructions are read. The index of the grammar code to be expanded is determined, and is placed atop the pushdown stack labeled PD. Several switches are set (SWITCH1, SWITCH2), to a neutral condition. If during the course of generation the specified level is exceeded, or if there are too many occurrences of any grammar code, they are set to a flagged state and dictionary expansion is forced wherever possible. The switch SWNR is set whenever expansion by a new rule is mandatory. The entries in the pushdown stack PD are of the form:

12 bits	48 bits
level of occurrence	grammar code index

GENUN performs generation using PD by looping on the condition of the stack pointer at the current node, named J. At the beginning of manufacturing a specific generation instance, a word of the form:

12 bits	48 bits
1	index (GC)

is placed in entry one of PD, the array serving as a pushdown stack. At each node J is either decremented (when lexical insertion occurs) or incremented (when rule expansion occurs). When J finally receives the value zero, the generation instance is complete, and the next example generation is started.

Suppose generation is at an entry I in the pushdown stack.

First G(1,J), the printout representation of grammar code PD position J is accessed and printed along with its level. Then several Boolean tests on the switches are made to determine if dictionary expansion is mandatory. If so, a random entry is selected using the directory information and subroutine RANDMOD (if, in fact, there are entries available). If not, then the number of rules is added to the number of dictionary entries, and a number between 0 and that sum minus one is selected using RANDMOD. If it is greater than the number of rules, a dictionary entry is determined by subtracting that sum from the number of rules and adding the result to the position of the first dictionary entry. The pointer to the top of the PD is decremented by 1 after dictionary expansion. If the random number is less than the number of rules, a phrase structure rule is selected by adding it to the position of the first rule. The rule will expand into several grammar codes; they are added to PD by the subroutine EXPAND, which also maintains proper bookkeeping on the condition of the stack.

As EXPAND is entered PD will be in the state

J →	level	index (GC)

(GC is the grammar code at the "top" of the stack.)

Both level and GC will have been printed. Suppose the rule selected was GC → GC1 + GC2 + GC3. (GC is constrained by SGP to expand into from one to four grammar codes.) In adapted form this rule will be in a 6400 word

zeroes	index (GC1)	index (GC2)	index (GC3)	index (GC)

This word is cyclically shifted by 12 bits to remove these indices in the order: index (GC3), index (GC2), index (GC1). The level is incremented so the Level' = Level + 1 and the level information is placed in the left 12 bits of words containing these indices. These words are successively placed on the top of the stack and J is incremented. After they are entered PD is in the state

old value of J →	Level'	index (GC3)
	Level'	index (GC2)
new stack pointer → J' = J + 2	Level'	index (GC1)

As these words are processed, a counter for each grammar code is incremented in the GC table, noting that GC1, GC2, and GC3 have appeared one more time. Control is then returned to GENUN (= the SGP).

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)*
University of California at Berkeley
2222 Piedmont Avenue
Berkeley, California 94720

2a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

2b. GROUP
N/A

3. REPORT TITLE
Research in Chinese-English Machine Translation

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*
Final 1 September 1968 - 31 August 1970

5. AUTHOR(S) *(First name, middle initial, last name)*
Dr. William S-Y Wang; Dr. Benjamin K. T'sou; Mr. Stephen W. Chan

6. REPORT DATE
November 1971

7a. TOTAL NO. OF PAGES
245

7b. NO. OF REFS
46

8a. CONTRACT OR GRANT NO.
F30602-69-C-0055

b. PROJECT NO.
Job Order 45940000

c.

d.

9a. ORIGINATOR'S REPORT NUMBER(S)
None

9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*
RADC-TR-71-211

10. DISTRIBUTION STATEMENT
Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY
Rome Air Development Center (IRDT
Griffiss AFB, New York 13440

13. ABSTRACT

The report documents results of a two-year R&D effort directed at designing a prototype system for Chinese-English machine translation in the general area of physics. Lexicography: Approximately 15,000 nuclear physics terms were added to the system's lexicon for a cumulative total of 57,000 entries. All entries were assigned grammar codes, romanization and English glosses. Compilation of lexical data was combined with the design of productive word derivation rules increasing the effective potential of the system's lexicon. Linguistics: Work on grammar ("Berkeley Grammar II") was focused on expansion and consolidation of syntactic recognition and parsing rules. Grammar codes in the lexicon and in the rules underwent a review for consistency, and redundancies were eliminated. New sets of rules were added as a result of processing Chinese texts in the field of nuclear physics. Testing and implementation of interlingual (Chinese-to-English) transfer rules concentrated on conversion of Chinese nominalizations and relativizations into English counterparts. Software: Testing of updated SAS programs and documentation of all programs were completed. In addition to continuing refinements of SAS programs, plotting routines were implemented in CalComp Plotter to output structural trees and plot corresponding sentences in Chinese characters.

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Machine Translation R&D						
Chinese-English Translation						
Automated Lexicography						
Computational Linguistics						
Computer Programming						
Input/Output Processing						